



**HEVs**

haute école valaisanne  
hochschule wallis



**HEVs2**

haute école valaisanne  
hochschule wallis

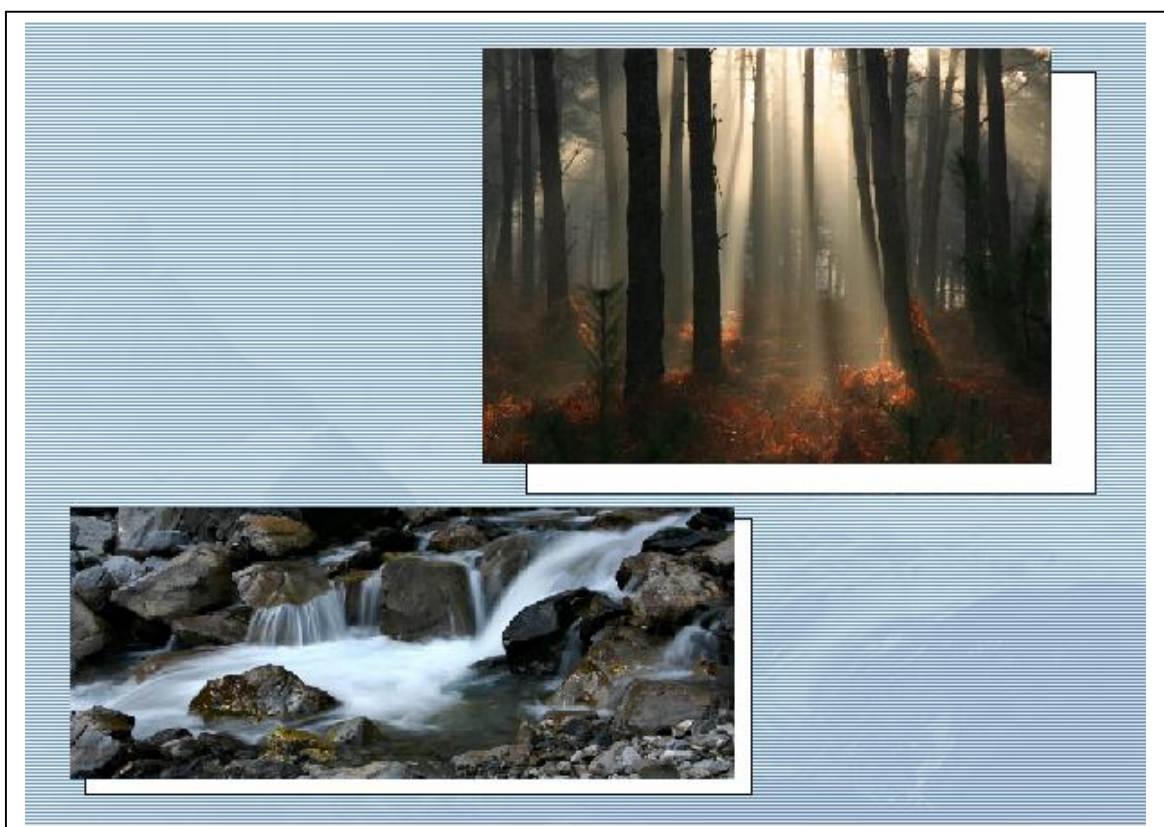
Filière informatique de gestion

**Diplôme 2006 / 2007**

Etudiante : Nathalie Solioz

Professeur : Yann Bocchi

## **Intégration des données spatiales dans les applications de gestion**



[www.hevs.ch](http://www.hevs.ch)

**Hes·SO**

Haute Ecole Spécialisée  
de Suisse occidentale  
Fachhochschule Westschweiz

# Table des matières

<b>INTRODUCTION.....</b>	<b>1</b>
Contexte du projet .....	1
Présentation du projet .....	1
Objectifs .....	1
<b>COMPARATIF DE L'ART DE LA DESCRIPTION DES DONNEES SPATIALES DANS LES SGBDR</b>	
<b>COURANTS.....</b>	<b>2</b>
Oracle .....	2
SQL Server .....	5
<i>MsSqlSpatial.....</i>	<i>5</i>
<i>SpatialWare .....</i>	<i>5</i>
<i>Comparatif SQL Server .....</i>	<i>5</i>
MySQL .....	7
PostGIS .....	7
Autres bases de données .....	10
<i>Access .....</i>	<i>10</i>
<i>DB2.....</i>	<i>10</i>
Choix de la solution .....	10
<b>SYSTEME D'INFORMATION GEOGRAPHIQUE (SIG-GIS) .....</b>	<b>11</b>
Les données spatiales .....	11
Format des données .....	12
<i>Raster .....</i>	<i>12</i>
<i>Vecteur.....</i>	<i>12</i>
<i>Représentation d'une zone géographique.....</i>	<i>13</i>
Les modèles de SIG.....	13
<i>Modèle spaghetti .....</i>	<i>13</i>
<i>Modèle topologique .....</i>	<i>14</i>
Les étapes des systèmes d'information géographique .....	15
<i>Abstraction.....</i>	<i>15</i>
<i>Acquisition.....</i>	<i>15</i>
<i>Archivage.....</i>	<i>15</i>
<i>Analyse.....</i>	<i>15</i>
<i>Affichage.....</i>	<i>15</i>
Les domaines d'utilisation des SIG.....	16
<b>COMPARATIF DE QUELQUES LOGICIELS SIG DU MARCHÉ.....</b>	<b>17</b>
ArcGIS .....	17
<i>Descriptif des outils ArcGIS.....</i>	<i>18</i>
MapInfo.....	19
<i>Descriptif des outils MapInfo.....</i>	<i>19</i>
Comparatif ArcView et MapInfo Professional.....	20
Oracle Application Server MapViewer .....	21
QGIS (Quantum GIS) .....	21
SharpMap .....	22

Récapitulatif .....	22
<b>OPEN GEOSPATIAL CONSORTIUM (OGC) .....</b>	<b>23</b>
<b>LA NORME OPENGIS : SIMPLE FEATURES SPECIFICATIONS FOR SQL .....</b>	<b>25</b>
Approche .....	25
Modèle Object Geometry .....	26
<i>Geometry</i> .....	26
<i>GeometryCollection</i> .....	27
<i>Point</i> .....	27
<i>MultiPoint</i> .....	27
<i>Curve</i> .....	27
<i>LineString, Line, LinearRing</i> .....	27
<i>MultiCurve</i> .....	28
<i>MultiLineString</i> .....	28
<i>Surface</i> .....	28
<i>Polygon</i> .....	29
<i>MultiSurface</i> .....	29
<i>MultiPolygon</i> .....	29
Relations entre les géométries .....	30
Fonctionnalités .....	30
<i>Fonction de base et de constructeur de géométrie</i> .....	30
<i>Point</i> .....	30
<i>Curve</i> .....	30
<i>LineString</i> .....	31
<i>Surface</i> .....	31
<i>Polygon</i> .....	31
<i>GeomCollection</i> .....	31
<i>Multicurve</i> .....	31
<i>MultiSurface</i> .....	31
<i>Opérateurs spatiaux</i> .....	31
La représentation binaire des géométries (WKB) .....	32
La représentation textuelle des géométries .....	33
La représentation textuelle des systèmes de référence spatiaux (WKT) .....	34
Implémentation des « feature tables » .....	35
Description des schémas .....	36
<i>Feature table</i> .....	36
<i>Table Geometry_Columns (métadonnées)</i> .....	37
<i>Table Spatial_Reference_System</i> .....	38
<i>Tables contenant les géométries (SQL92 seulement)</i> .....	38
<b>UTILISATION D'ORACLE SPATIAL .....</b>	<b>41</b>
Oracle Spatial .....	41
<i>Systèmes de coordonnées</i> .....	41
<i>Types de coordonnées</i> .....	42
<i>Sdo_tolerance et arc_tolerance</i> .....	42
Création de données géométriques .....	43
<i>Création de la table</i> .....	43
<i>Création des métadonnées</i> .....	43
<i>Création des index</i> .....	44
<i>Insertion de données</i> .....	44
Index spatial basé sur des fonctions .....	46

Utilisation de fonctions Oracle Spatial.....	47
<i>Buffer</i> .....	47
<i>Filtre</i> .....	48
<i>Coordonnées</i> .....	49
<i>Intersection</i> .....	49
Utilisation d'Oracle et de PL/SQL .....	50
<i>Procédure stockée</i> .....	50
<b>PROCESSUS DE RECHERCHE D'ITINERAIRES.....</b>	<b>53</b>
1° Choix du lieu de départ.....	53
2° Choix du rayon .....	53
3° Choix du type d'activité et du type de route.....	53
4° Longueur du trajet .....	54
<i>Exemple</i> .....	54
5° Type de parcours : boucle – aller simple – aller-retour .....	54
6° Calcul du dénivelé, de la difficulté et de la durée du trajet .....	55
<i>Exemple</i> .....	55
Adaptation .....	56
<b>GRAPHE .....</b>	<b>57</b>
Tables .....	57
<i>Table de conversion</i> .....	57
<i>Table de départ</i> .....	58
<i>Table d'arrivée</i> .....	58
<i>Table des distances</i> .....	58
<i>Tableau des points de départ</i> .....	58
<i>Tableau des liens</i> .....	59
<i>Tableau des intersections</i> .....	59
Graphe .....	60
<i>Généralité</i> .....	60
<i>Création du graphe</i> .....	60
<i>Recherche d'itinéraire</i> .....	62
<b>INTERFACE « EXPORTATION » .....</b>	<b>67</b>
<b>API GOOGLEMAPS .....</b>	<b>68</b>
Fichier de base .....	68
Affichage d'une carte .....	69
Poly Ligne .....	70
Marqueurs (balises) .....	70
Marqueurs personnalisés .....	71
<b>GPS.....</b>	<b>72</b>
Fonctionnement d'un GPS .....	72
Fonctions embarquées sur les récepteurs GPS .....	72
Format .....	73
<i>GPX</i> .....	73
<i>PCX5</i> .....	74
<i>NMEA 0183</i> .....	75

TXT .....	75
<b>INTEGRATION DES DONNEES .....</b>	<b>76</b>
Strada-DB.....	76
<i>Présentation</i> .....	76
<i>Structure des données</i> .....	76
<i>Description des géométries</i> .....	77
Données des mensurations cadastrales .....	78
<i>Présentation</i> .....	78
<i>Structure des données</i> .....	78
<i>Problèmes</i> .....	79
Scénario .....	80
<b>GESTION DE PROJET .....</b>	<b>81</b>
Cahier des charges.....	81
<i>Contexte du projet</i> .....	81
<i>Présentation du projet</i> .....	81
<i>Objectifs</i> .....	81
Découpage du travail.....	83
Planification .....	84
<i>Initiale</i> .....	84
<i>Modification</i> .....	84
Déroulement du projet.....	86
<b>SYNTHESE .....</b>	<b>88</b>
<b>REMERCIEMENTS .....</b>	<b>89</b>
<b>SOURCES .....</b>	<b>90</b>
Bibliographie .....	90
<i>Comparatif</i> .....	90
<i>Norme</i> .....	90
<i>GIS – SIG</i> .....	90
<i>GPS</i> .....	91
<i>Intégration des données</i> .....	91
<i>Oracle Spatial</i> .....	91
<i>ChartFX</i> .....	91
URL .....	92
<i>Comparatif</i> .....	92
<i>Norme</i> .....	92
<i>GPS</i> .....	93
<i>API Google Maps</i> .....	93
<i>PL/SQL</i> .....	93
<i>Oracle Spatial</i> .....	93
Outils .....	94
<i>WebApplication</i> .....	94
<i>ShapeViewer</i> .....	94
<i>Shp2Sdo</i> .....	94
<i>ChartFX</i> .....	95
<i>Framework .Net 2.0</i> .....	96
<b>ANNEXES.....</b>	<b>97</b>
Use cases .....	97

Acteurs.....	97
Accès .....	97
Fonctionnalités du site .....	98
Modèle physique de données .....	101
Tables de données classiques .....	102
Tables de données spatiales.....	103
Tables mdsys.....	103
Mise en place d'une image VMWare .....	104
Pré requis.....	104
Installation du LoopBack Adaptor .....	104
Installation d'Oracle .....	104
Configuration d'Oracle .....	105
Installation du site web .....	105
Configuration du site web .....	106
Description de l'application web .....	108
Page d'accueil .....	108
Recherche « Départ – Arrivée » .....	109
Recherche « Départ » .....	110
Détails .....	112
Création de géométries : Exemples .....	114
Introduction .....	114
Point.....	114
Cluster de points.....	114
Ligne droite .....	115
Arc de cercle.....	115
Ligne composée .....	115
Polygone simple avec ligne droite .....	116
Polygone simple avec arc.....	116
Polygone particulier : rectangle .....	117
Polygone particulier : cercle.....	117
Polygone composé simple (avec trou et ligne droite).....	118
Polygone composé complexe (avec trou, ligne droite et arc de cercle).....	118
Fonctions et procédures Oracle développées .....	119
Fonctions .....	119
Package .....	123
Requêtes Visual Studio .....	125
Problèmes rencontrés .....	127
Oracle Spatial .....	127
Visual Studio.....	130
Graphe.....	131
IIS - Oracle.....	134
Personnes de contact.....	136
Contenu du DVD .....	137
Attestation .....	138

# Introduction

## Contexte du projet

---

Ce projet a été réalisé dans le cadre du travail de diplôme de la formation d'informaticien de gestion HES à l'HEVs de Sierre. Ce travail s'est déroulé sur une période de 12 semaines, du 25 septembre 2006 au 18 décembre 2006.

## Présentation du projet

---

De plus en plus d'applications de gestion sont appelées à intégrer des données spatiales liées à du positionnement géographique ou à des itinéraires. Les domaines d'activités touchés sont très variés : gestion de l'aménagement du territoire, agriculture, prévention des risques naturels, gestion des infrastructures routières et médecine ne sont qu'une infime part ces domaines.

La centralisation des données géo-spatiales dans une base de données permet de garder une universalité et une intégrité des données ainsi qu'une mise à jour fréquente de celles-ci, ce qui n'était pas envisageable dans le passé avec de simples cartes.

## Objectifs

---

L'objectif de ce projet est tout d'abord d'explorer les fonctionnalités GIS (Geographical Information Systems ou Système d'Informations Géographiques en français) offertes par les bases de données. Un système d'informations géographiques constitue l'ensemble du processus visant à la gestion des données géo-spatiales : acquisition, archivage, abstraction, analyse et affichage des données (5A). Le but de ce travail n'est cependant pas de gérer l'intégralité d'un GIS, mais de se concentrer plus particulièrement sur la partie analyse et l'utilisation des données et fonctions du GIS dans une application d'informatique de gestion.

Le second objectif est donc l'analyse détaillée de la structure des données géo-spatiales ainsi que des fonctions/algorithmes existants/à développer pour exploiter ces informations.

Finalement, les informations spatiales doivent pouvoir être gérées avec des données de gestion classique.

## Comparatif de l'art de la description des données spatiales dans les SGBDR courants

Les différentes librairies et programmes présentés dans ce comparatif ne sont pas destinés à des utilisateurs finaux mais à des développeurs. Il est important de noter que la plupart des bases de données offrent une extension spatiale mais que celle-ci ne permet pas, en général, d'afficher une représentation graphique des résultats.

Toutes les bases de données de ce comparatif suivent plus ou moins la norme « Simple Features Specifications for SQL », mise en place par l'Open Geospatial Consortium (OGC). Cette norme sera décrite plus en détail dans le chapitre « la norme OpenGIS : Simple Features Specifications for SQL ».

### Oracle <sup>1</sup>

Oracle propose deux extensions à sa base de données (version 10g) pour la gestion des données spatiales : Oracle Spatial et Oracle Locator.

Oracle Locator est inclus dans Oracle Entreprise Edition, Standard Edition et Standard Edition One. Il n'a qu'une partie limitée des fonctions et procédures de Oracle Spatial mais il possède les mêmes types de données, opérateurs et index. Cette version n'est pas idéale pour des SIG qui requièrent un management complexe des données spatiales, comme le support de GeoRaster ou des modèles de données topologiques et réseaux.

Oracle Spatial	Oracle Locator
Type de données propre à Oracle : sdo_geometry. Elle permet de gérer toutes les géométries (Figure 1)	Idem
Formats WKT et WKB	Idem
Indexation des données avec des R-Tree, pour améliorer les performances d'accès	Idem
Utilisation de requêtes SQL pour créer, modifier et supprimer des index	Idem
Index spatiaux partitionnés : permet d'améliorer le temps d'accès aux données et permet de faire des requêtes parallèles sur des recherches sur plusieurs partition	Non
Partitionnement de table pour le support d'index spatiaux	Idem *
Réplication d'objets géométriques	Idem
Réplication multiple de géométrie	Idem *
Transformation implicite des coordonnées	Idem
Supporte les données 2D, 3D et 4D	Idem
Requêtes two-tiered : Filtre les données d'abord sur l'élément de la géométrie qui satisfait à la condition puis sur les relations (ex : recherche d'abord les clients dans les 5 km d'un magasin puis ceux qui ont dépensés plus de 5000.- cette année)	Idem

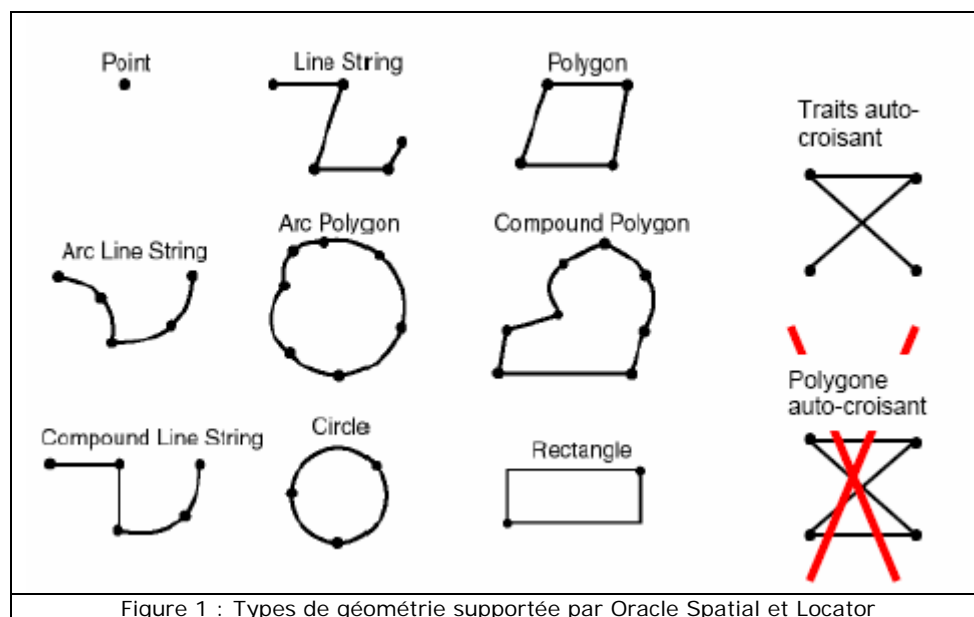
<sup>1</sup> Oracle Spatial User's Guide and Reference 10g Release 2  
<http://www.oracle.com/technology/documentation/spatial.html>  
 Comparison of the Oracle Spatial and Locator features  
<http://www.mattpopper.com/?p=26>



<p>Opérateurs spatiaux :</p> <ul style="list-style-type: none"> <li>- sdo_filter</li> <li>- sdo_join</li> <li>- sdo_nn</li> <li>- sdo_nn_distance</li> <li>- sdo_relate</li> <li>- sdo_within_distance</li> <li>- sdo_anyinteract</li> <li>- sdo_contains</li> <li>- sdo_coveredby</li> <li>- sdo_covers</li> <li>- sdo_equal</li> <li>- sdo_inside</li> <li>- sdo_on</li> <li>- sdo_overlapbdydisjoint</li> <li>- sdo_overlapbdyintersect</li> <li>- sdo_overlaps</li> <li>- sdo_touch</li> </ul>	Idem
<p>Fonctions d'agrégation :</p> <ul style="list-style-type: none"> <li>- sdo_aggr_centroid</li> <li>- sdo_aggr_concat_lines</li> <li>- sdo_aggr_convexhull</li> <li>- sdo_aggr_lrs_concat</li> <li>- sdo_aggr_mbr</li> <li>- sdo_aggr_union</li> </ul>	<ul style="list-style-type: none"> <li>- sdo_aggr_mbr : fonction qui retourne le rectangle minimum entourant la géométrie</li> </ul>
<p>Package Geometry (sdo_geom) :</p> <ul style="list-style-type: none"> <li>- fonctions de relation entre géométrie <ul style="list-style-type: none"> <li>o relate</li> <li>o within_distance</li> </ul> </li> <li>- fonctions de validation <ul style="list-style-type: none"> <li>o validate_geometry_with_context</li> <li>o validate_layer_with_context</li> </ul> </li> <li>- fonctions sur un objet <ul style="list-style-type: none"> <li>o sdo_arc_densify</li> <li>o sdo_area</li> <li>o sdo_buffer</li> <li>o sdo_centroid</li> <li>o sdo_convexhull</li> <li>o sdo_length</li> <li>o sdo_max_mbr_ordinate</li> <li>o sdo_min_mbr_ordinate</li> <li>o sdo_mbr</li> <li>o sdo_pointonsurface</li> </ul> </li> <li>- fonctions sur deux objets <ul style="list-style-type: none"> <li>o sdo_distance</li> <li>o sdo_difference</li> <li>o sdo_intersection</li> <li>o sdo_union</li> <li>o sdo_xor</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- fonctions sur deux objets <ul style="list-style-type: none"> <li>o sdo_distance</li> </ul> </li> <li>- fonctions de validation <ul style="list-style-type: none"> <li>o sdo_geom.validate_geometry_with_context</li> <li>o sdo_geom.validate_layer_with_context</li> </ul> </li> </ul>
<p>Package Geocoding (sdo_GCDR) : lien entre longitude/latitude et adresse postale</p>	non
<p>Package Linear Referencing System (LRS) : Les informations de "mesures" liées à une géométrie linéaire sont stockées avec la géométrie et non séparément. Pas besoin de dupliquer les géométries</p>	non
<p>Package d'utilitaire spatial (sdo_util)</p>	Idem

Package de tuning (sdo_tune)	Idem
Package Migration (sdo_migrate) : permet de mettre à jour les données d'anciennes versions	Idem
Package d'analyse spatiale et de data-mining (sdo_sam) :	Non
Package de transformation des coordonnées (sdo_cs) : Permet de transformer explicitement ou implicitement les géométries et les couches en longitude/latitude	<ul style="list-style-type: none"> <li>- sdo_cs.transform</li> <li>- sdo_cs.transform_layer</li> </ul>
Supporte GeoRaster → gestion des données raster	Non
Modèle de données topologiques : utilisé quand beaucoup de données sont modifiées et qu'il est nécessaire d'avoir une intégrité des données entre les cartes et les couches.	Non
Modèle de données réseau (routing) Ex : Puis-je faire un virage à droite à cette intersection. Où est la prochaine gare	Non
Supporte données géodésiques	Idem
Construction d'index spatiaux parallèles	Idem *
Gratuit	Gratuit

\* pas supporté sur Oracle Standard Edition mais seulement sur Entreprise Edition



## SQL Server

### MsSqlSpatial

CodePlex, le site web de la communauté de développement de Microsoft, propose gratuitement une extension spatiale pour SQL Server 2005 (Express, Workgroup, Standard et Entreprise Edition) appelée MsSQLSpatial<sup>2</sup>.

### SpatialWare

MapInfo propose une extension spatiale appelée SpatialWare<sup>3</sup>. Cette librairie n'est pas gratuite, mais cependant, une démonstration de 60 jours est disponible en téléchargement<sup>4</sup> sur le site de MapInfo.

## Comparatif SQL Server

MsSqlSpatial	SpatialWare
Type de données <ul style="list-style-type: none"> <li>- Point</li> <li>- LineString</li> <li>- Polygon</li> <li>- MultiPoint</li> <li>- MultiLineString</li> <li>- MultiPolygin</li> <li>- GeometryCollection</li> </ul>	<ul style="list-style-type: none"> <li>- St_spatial : type de données définies par l'utilisateur dans SQL Server</li> </ul>
Fonctions basiques : Permet de savoir où se trouve géographiquement une géométrie par rapport à une autre <ul style="list-style-type: none"> <li>- Equals</li> <li>- Disjoint</li> <li>- Intersects</li> <li>- Touches</li> <li>- Crosses</li> <li>- Within</li> <li>- Contains</li> <li>- Overlaps</li> <li>- Covers</li> <li>- CoveredBy</li> <li>- Relate</li> <li>- IsWithinDistance</li> </ul>	Même fonctionnalités présentes mais ajout d'une série de fonctions supplémentaires : <ul style="list-style-type: none"> <li>- HG_At_End_Of</li> <li>- HG_At_Start_Of</li> <li>- HG_Connected_To</li> <li>- HG_Is_Contiguous</li> <li>- ...</li> </ul>
Fonctions d'analyse spatiale <ul style="list-style-type: none"> <li>- ConvexHull</li> <li>- Intersection</li> <li>- Union</li> <li>- Difference</li> <li>- SymDifference</li> <li>- IsValid</li> <li>- IsRing</li> <li>- IsSimple</li> <li>- ...</li> </ul>	Idem. Ajout d'une série de fonctions comme : <ul style="list-style-type: none"> <li>- HG_Affine</li> <li>- HG_Center_In</li> <li>- HG_Erase</li> <li>- HG_SphericalBuffer</li> <li>- HG_SphericalCircle</li> <li>- ...</li> </ul>

<sup>2</sup> <http://www.codeplex.com/Wiki/View.aspx?ProjectName=MsSqlSpatial>

<sup>3</sup> SpatialWare User Guide

<http://extranet.mapinfo.com/support/documentation/manuals.cfm>

<sup>4</sup> <http://extranet.mapinfo.com/products/Download.cfm?ProductID=1141&productcategoryid=>

Fonctions de mesure <ul style="list-style-type: none"> <li>- Distance</li> <li>- Length</li> <li>- Area</li> <li>- Buffer</li> <li>- Centroid</li> <li>- PointOnSurface</li> <li>- Simplify</li> <li>- ...</li> </ul>	Idem. Même fonctionnalisées plus ajout de fonctions comme : <ul style="list-style-type: none"> <li>- HG_Azimuth</li> <li>- HG_Height</li> <li>- HG_Separation</li> <li>- HG_Slope</li> <li>- HG_SphericalArea</li> <li>- HG_Width</li> <li>- ST_Length_3D</li> <li>- ST_Perimeter</li> <li>- ...</li> </ul>
Fonctions d'agrégation <ul style="list-style-type: none"> <li>- UnionAggregate</li> <li>- EnvelopeAggregate</li> <li>- IntersectionAggregate</li> <li>- Collect</li> <li>- MakeLine</li> <li>- Polygonize</li> </ul>	<ul style="list-style-type: none"> <li>- HG_Aggspatial</li> <li>- HG_Aggunion</li> <li>- HG_Aggintersection</li> <li>- HG_Aggconvex_Hull</li> <li>- HG_Bounding_box</li> </ul>
Fonctions de casting : Permet de créer une géométrie par rapport à du texte ou à un WKB (binaire) ou l'inverse. <ul style="list-style-type: none"> <li>- AsBinary</li> <li>- AsText</li> <li>- GeomFromText</li> <li>- GeomFromWKB</li> <li>- PointFromText</li> <li>- ... (2 fonctions par géométrie)</li> </ul>	Idem. Seulement, comme SpatialWare utilise sa propre géométrie, il n'a pas été nécessaire de créer une fonction par type de géométrie. Une seule suffit pour les gérer toutes. <ul style="list-style-type: none"> <li>- HG_AsBinary</li> <li>- HG_AsText</li> <li>- HG_GeometryFromBinary</li> <li>- HG_GeometryFromText</li> </ul>
Fonctions sur les coordonnées : Permet de transformer une géométrie d'un système de coordonnées à un autre. <ul style="list-style-type: none"> <li>- Transform</li> </ul>	Idem <ul style="list-style-type: none"> <li>- HG_CSTransform</li> </ul>
non	Fonctions générales <ul style="list-style-type: none"> <li>- HG_GetString</li> <li>- HG_Version</li> </ul>
Fonctions d'observation <ul style="list-style-type: none"> <li>- X</li> <li>- Y</li> <li>- EndPoint</li> <li>- StartPoint</li> <li>- PointN</li> <li>- GeometryN</li> <li>- GeometryType</li> </ul>	Idem. Les mêmes fonctions sont présentes mais ajout d'une série de fonctions, telles que : <ul style="list-style-type: none"> <li>- HG_Ori_X</li> <li>- HG_Pointdyn_Orix_X</li> <li>- HG_Urt_X</li> <li>- HG_Start_Tangent</li> <li>- HG_Radians</li> <li>- HG_Cen_X</li> <li>- ...</li> </ul>
Fonction AsGML	Non
Format <ul style="list-style-type: none"> <li>- WKT</li> <li>- WKB</li> </ul>	<ul style="list-style-type: none"> <li>- WKT</li> <li>- WKB (binaire)</li> </ul>
Importation de shapefiles et de tables PostGIS	Pour importer des données, il faut utiliser un autre programme, comme, par exemple, EasyLoader de MapInfo

Indexation spatiale simple <ul style="list-style-type: none"> <li>- FilterQuery</li> <li>- JoinFilterQuery</li> <li>- IsWithinDistanceQuery</li> <li>- RelateQuery</li> </ul>	Indexation spatiale grâce à des R-Trees
Pas de fonctions spéciales pour la 3D	Possède des fonctions spéciales pour la 3 dimension
Conforme à la norme Features Specification for SQL Revision 1.1 (pas officiel mais vérifiable)	OGC naming and parameter specifications for spatial functions in SQL.
Systèmes de coordonnées <ul style="list-style-type: none"> <li>- SRID</li> <li>- SetSRID</li> <li>- Transform</li> </ul>	- SRID
Gratuit	Payant (Prix inconnu)

## MySQL<sup>5</sup>

MySQL inclus automatiquement dans sa base de données l'extension spatiale. MySQL se base sur la norme OpenGIS mais toutes les fonctionnalités n'ont pas été mises en place. Voici un tableau présentant les fonctionnalités implémentées et celles que ne le sont pas :

Implémenté	Non implémenté
Type de données : <ul style="list-style-type: none"> <li>- Point</li> <li>- Curve</li> <li>- LineString</li> <li>- Surface</li> <li>- Polygon</li> <li>- GeometryCollection</li> <li>- MultiPoint</li> <li>- MultiCurve</li> <li>- MultiLineString</li> <li>- MultiSurface</li> <li>- MultiPolygon</li> </ul>	
Format : <ul style="list-style-type: none"> <li>- WKT</li> <li>- WKB</li> </ul>	
Opérateurs spatiaux	
Indexation des données	
Fonction WKT	<ul style="list-style-type: none"> <li>- BdPolyFromText</li> <li>- BdMPolyFromText</li> </ul>
Fonction WKB	<ul style="list-style-type: none"> <li>- BdPolyFromWKB</li> <li>- BdMPolyFromWKB</li> </ul>
Fonction d'analyse	<ul style="list-style-type: none"> <li>- Boundary</li> <li>- IsEmpty</li> <li>- IsSimple</li> <li>- IsRing</li> <li>- Centroid</li> <li>- PointOnSurface</li> </ul>

<sup>5</sup> <http://forum.mysql.com/doc/refman/5.0/fr/spatial-extensions.html>

	Opérateurs géométriques : <ul style="list-style-type: none"> <li>- Intersection</li> <li>- Union</li> <li>- Difference</li> <li>- SymDifference</li> <li>- Buffer</li> <li>- ConvexHull</li> </ul>
	SRID
	Vues de méta-données

## PostGIS

PostGIS est le module spatial qui permet à PostgreSQL d'être considéré comme une base de données spatiale. PostGIS est, depuis début septembre 2006, conforme au niveau des types et des fonctions, à la norme OpenGIS Simple Features Specification for SQL. PostGIS<sup>6</sup> a cependant mis en place des fonctionnalités supplémentaires non définies par la norme. Le tableau ci-dessous en fait mention :

Conforme OpenGIS	Implémenté par PostGIS
Type de géométrie <ul style="list-style-type: none"> <li>- Point</li> <li>- Line</li> <li>- Polygon</li> <li>- MultiPoint</li> <li>- MultiLine</li> <li>- MultiPolygon</li> <li>- GeometryCollections</li> </ul>	
Format <ul style="list-style-type: none"> <li>- WKT</li> <li>- WKB</li> </ul>	<ul style="list-style-type: none"> <li>- EWKT</li> <li>- EWKB</li> </ul>
Gestion des index <ul style="list-style-type: none"> <li>- R-Tree</li> </ul>	<ul style="list-style-type: none"> <li>- GiST</li> <li>- B-Tree</li> </ul>
Fonctions de management <ul style="list-style-type: none"> <li>- AddGeometryColumn</li> <li>- DropGeometryColumn</li> <li>- SetSRID</li> </ul>	
Fonctions de relation entre géométrie <ul style="list-style-type: none"> <li>- Distance</li> <li>- Equals</li> <li>- Disjoint</li> <li>- Intersects</li> <li>- Touches</li> <li>- Crosses</li> <li>- Within</li> <li>- Overlaps</li> <li>- Contains</li> <li>- Relate</li> </ul>	
Fonctions sur les géométries <ul style="list-style-type: none"> <li>- Centroid</li> <li>- Area</li> <li>- Length</li> <li>- PointOnSurface</li> <li>- Boundary</li> <li>- Buffer</li> </ul>	

<sup>6</sup> <http://postgis.refrations.net/docs/>

<ul style="list-style-type: none"> <li>- ConvexHull</li> <li>- Intersection</li> <li>- SymDifference</li> <li>- Difference</li> <li>- GeomUnion</li> <li>- MemGeomUnion</li> </ul>	
<p>Accès aux géométries</p> <ul style="list-style-type: none"> <li>- AsText</li> <li>- AsBinary</li> <li>- SRID</li> <li>- Dimension</li> <li>- Envelope</li> <li>- IsEmpty</li> <li>- IsSimple</li> <li>- IsClosed</li> <li>- IsRing</li> <li>- NumGeometries</li> <li>- GeometryN</li> <li>- NumPoints</li> <li>- PointN</li> <li>- ExteriorRing</li> <li>- NumInteriorRings</li> <li>- InteriorRingN</li> <li>- EndPoint</li> <li>- StartPoint</li> <li>- GeometryType</li> <li>- X</li> <li>- Y</li> <li>- Z</li> <li>- M</li> </ul>	
<p>Constructeurs</p> <ul style="list-style-type: none"> <li>- GeomFromText</li> <li>- PointFromText</li> <li>- LineFromText</li> <li>- LineStringFromText</li> <li>- PolyFromText</li> <li>- PolygonFromText</li> <li>- MPointFromText</li> <li>- MLineFromText</li> <li>- MPolyFromText</li> <li>- GeomCollFromText</li> <li>- GeomFromWKB</li> <li>- PointFromWKB</li> <li>- LineFromWKB</li> <li>- LineStringFromWKB</li> <li>- PolyFromWKB</li> <li>- PolygonFromWKB</li> <li>- MPointFromWKB</li> <li>- MLineFromWKB</li> <li>- MPolyFromWKB</li> <li>- GeomCollFromWKB</li> <li>- BdPolyFromText</li> <li>- BdMPolyFromText</li> </ul>	
	<p>Extensions PostGIS</p> <ul style="list-style-type: none"> <li>- Fonctions de management</li> <li>- Opérateurs</li> <li>- Fonctions de mesure</li> </ul>

	<ul style="list-style-type: none"><li>- Type de sortie de géométries</li><li>- Constructeur de géométries</li><li>- Editeurs de géométries</li><li>- Référencement linéaire</li><li>- Autres...</li></ul>
--	---

## Autres bases de données

---

### Access

Ms Access n'a pas, à proprement parler, d'extension spatiale. Par contre, divers logiciels SIG ont les capacités de se connecter à une base de données Access pour afficher des données géo-spatiales. L'outil le plus souvent utilisé est Geomedia, mais Access est également compatible avec le logiciel ArcGIS d'ESRI.

### DB2

IBM propose pour sa base de données DB2 une extension spatiale appelée Spatial Extender<sup>7</sup>. Cette extension est conforme à la norme Simple Features Specifications pour SQL d'OpenGis et à ISO SQL/MM. Il peut être pris en charge par différents logiciels SIG comme ArcView d'ESRI. Tout comme pour les bases de données présentées précédemment, Spatial Extender permet de gérer des données géo-spatiales. De nouveaux types de données (géométries) sont créés sur DB2 Universal Database ainsi que de nouvelles fonctions et procédures.

Cette extension n'est cependant pas gratuite. On peut l'acheter en ligne pour 112.21 Euro sur le site français d'IBM.

## Choix de la solution

---

Des tests effectués avec MsSQLSpatial et Oracle Spatial ont permis de faire un choix pour le développement du prototype. Comme les deux solutions suivent la norme Simple Features Specification for SQL et se ressemblent beaucoup, il a fallu regarder plus en détail leurs différences. Au final, il en est ressorti que la meilleure solution pour gérer le site à réaliser serait d'utiliser Oracle Spatial. En effet, la présence d'une coordonnée Z (altitude) dans la gestion des géométries, permet d'avoir plus de précision dans l'analyse des informations. L'altitude étant un paramètre essentiel dans la gestion des données spatiales, le choix a donc rapidement été pris.

---

<sup>7</sup> [www-306.ibm.com/software/info/ecatalog/fr\\_FR/products/C105970D58974G08.html](http://www-306.ibm.com/software/info/ecatalog/fr_FR/products/C105970D58974G08.html)



## Système d'information géographique (SIG-GIS)

Un système d'information géographique est un ensemble de matériels, de logiciels et de processus mis en place pour permettre la collecte, la gestion, la manipulation, l'analyse, la modélisation et l'affichage de données numériques spatiales. Ces informations ont une référence au territoire et sont représentées par des coordonnées (longitude/latitude), un nom ou une adresse.

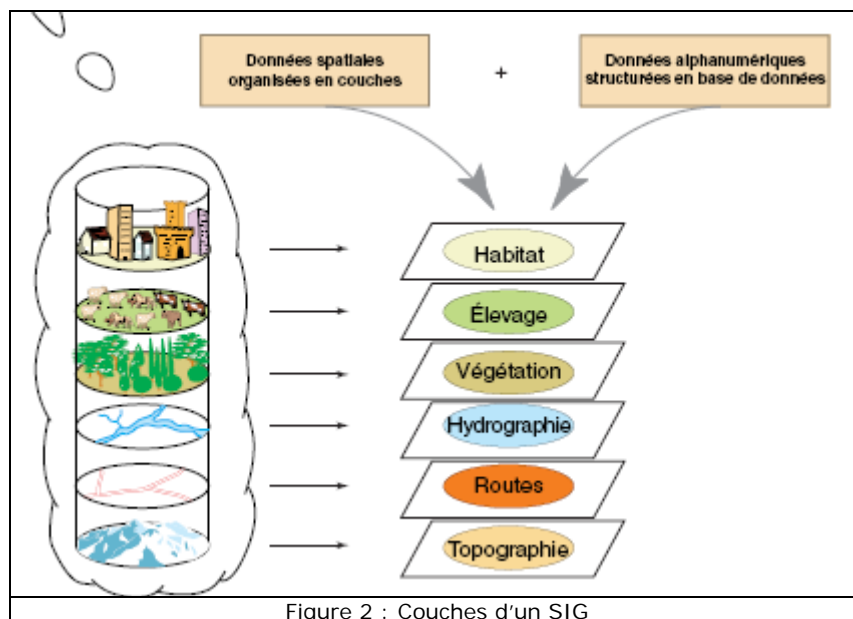
Le but d'un SIG est de répondre à 5 questions fondamentales :

- où : où se trouve la zone que je veux analyser
- quoi : que trouve-t-on sur l'emplacement analysé
- comment : comment les objets de cette zone sont-ils répartis et comment interagissent-ils entre eux
- quand : y a-t-il eu, à un moment donné, des changements sur les données
- et si : que se passerait-il si tel événement se produisait

Pour pouvoir répondre à ces questions, il a fallu mettre en place une méthode pour stocker les informations relatives à la géographie.

### Les données spatiales

Un SIG est un outil permettant de représenter et d'analyser toutes les choses qui existent sur terre. Comme le montre l'image ci-dessous, une zone géographique sera décortiquée en couches thématiques, comme la topographie, l'hydrographie ou la végétation. Chaque couche sera représentée dans la base de données par sa propre table.



Une donnée spatiale est aussi, dans la plupart des cas, liée à des données alphanumériques qui ajoutent des informations textuelles, qualitatives ou quantitatives aux données spatiales.

Ces données alphanumériques permettent d'avoir des informations démographiques, économiques, administratives, sociales,... supplémentaires et permettent également de mettre en évidence un côté temporel, pour savoir, par exemple, à quel moment, des changements sont intervenus ou quel est l'âge de tel objet,...

Cependant, ces tables nous permettent seulement d'avoir une représentation textuelle ou sémantique de la géographie, alors que l'intérêt d'un SIG est la représentation sur carte de ces informations. C'est pourquoi deux formats ont été mis en place pour représenter ces données spatiales.

## Format des données

### Raster

Une donnée raster est une donnée représentant le monde selon un maillage défini, régulier. Chaque cellule (pixel) possède une ou plusieurs valeurs décrivant les caractéristiques de l'espace. Ainsi, chaque point du territoire possède une information détaillée de sa géographie.

Ce format est utilisé avec des images provenant de photos aériennes, de cartes scannées, d'images satellites ou d'images radars.

Ce format nécessite donc beaucoup de mémoire et a quelques difficultés à représenter des faits géographiques linaires.

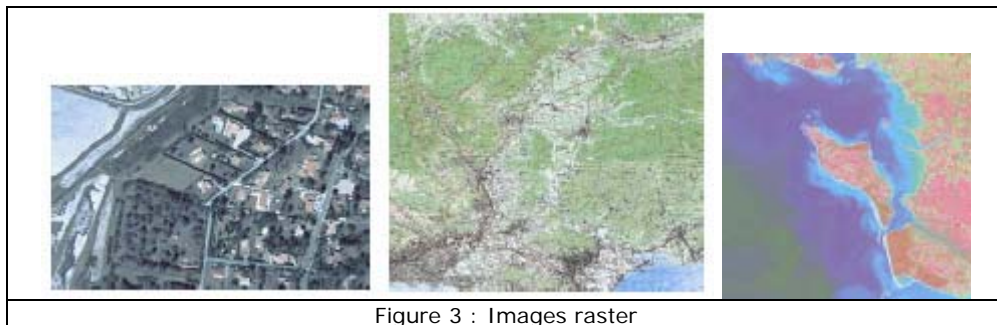


Figure 3 : Images raster

### Vecteur

La donnée vecteur représente chaque objet d'une zone géographique à partir de points, lignes ou surfaces, définis par des coordonnées réelles (x, y).

Ainsi, selon l'échelle utilisée, une ville pourra être définie soit par un point (grande échelle), soit par une surface (petite échelle).

Une ligne pourra représenter l'hydrographie, les axes routiers, des réseaux de communications. Alors qu'une surface matérialisera une entité abstraite comme la surface d'une commune ou alors une forêt, un lac ou une zone d'habitation.

Toutes ces informations pourront être liées entre elles pour former des objets plus complexes. Une ville est représentée par une surface, mais possède aussi un réseau routier, des zones d'habitation, des maisons, des arbres,...

Ce format demande beaucoup moins de mémoire pour gérer les données.

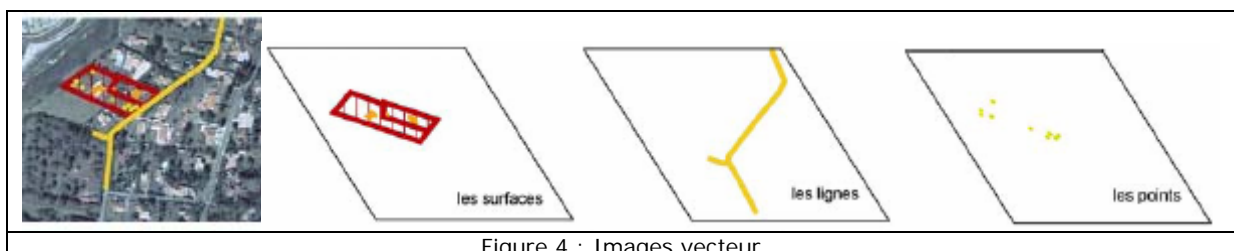


Figure 4 : Images vecteur

## Représentation d'une zone géographique

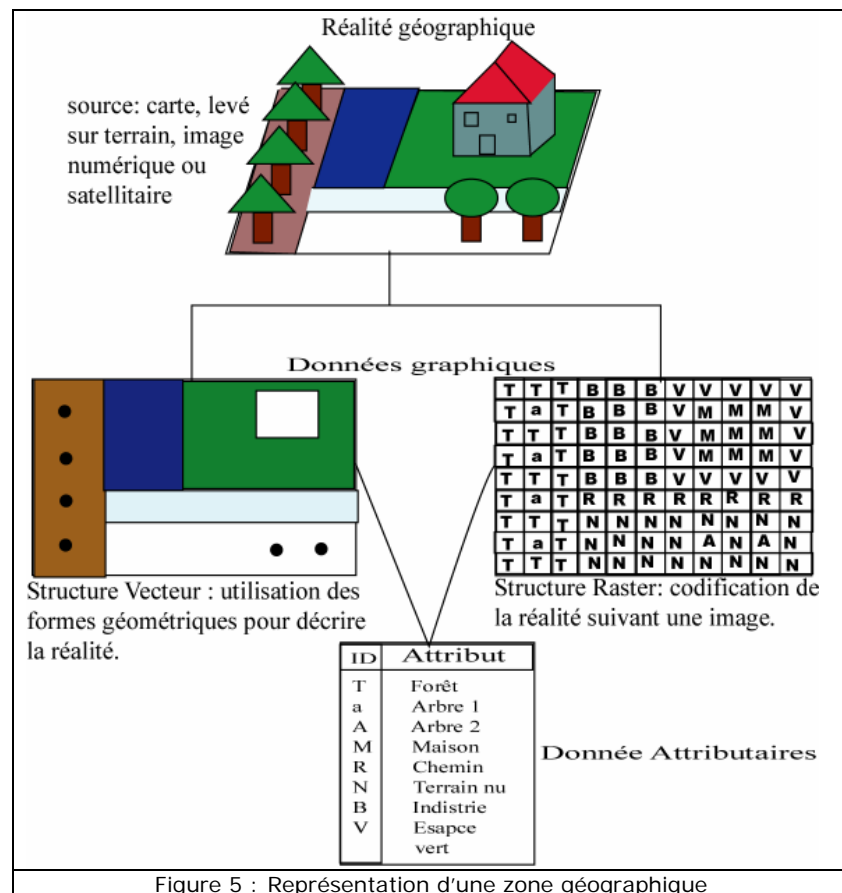


Figure 5 : Représentation d'une zone géographique

## Les modèles de SIG

Un modèle permet d'analyser la réalité et de la schématiser pour rendre les données récoltées utilisables. Deux modèles existent, le modèle spaghetti et le modèle topologique.

### Modèle spaghetti

Dans un modèle spaghetti, les données sont décrites séparément les unes des autres. Aucune relation entre les différents segments n'est précisée. Il y aura donc des informations redondantes pour permettre de lier les informations ensemble. Ce genre de structure ne sert pratiquement qu'à une représentation visuelle des données. Il est très difficile d'effectuer de l'analyse spatiale à partir de ce type de modèle.

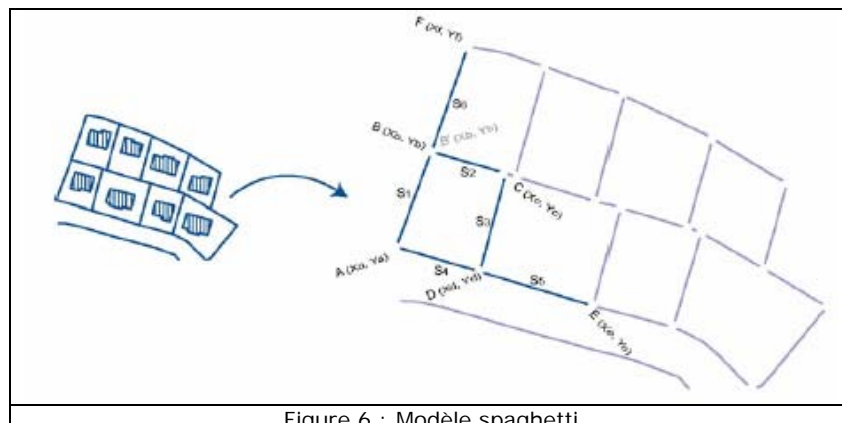


Figure 6 : Modèle spaghetti

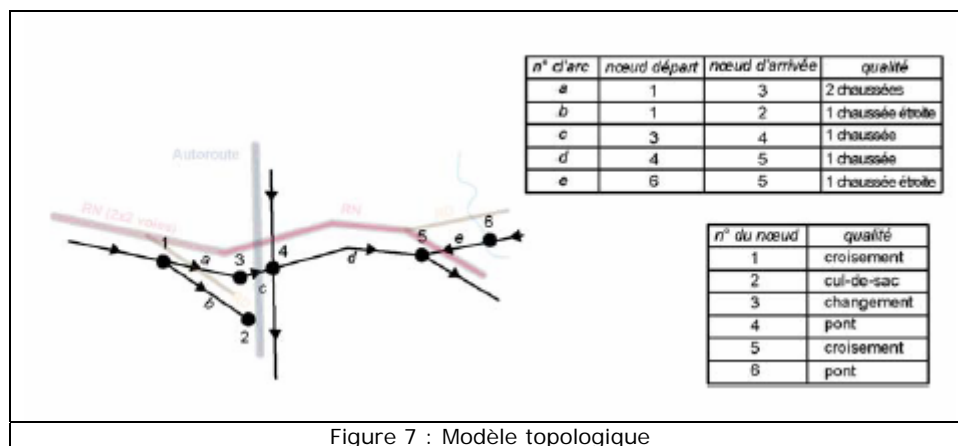
L'image ci-dessus représente un modèle spaghetti. Pour décrire dans la base de données l'image de gauche, il faut décrire chaque segment. Ainsi, le segment S1 à pour sommet A et B. Le segment S2 a pour sommet B et C.

On voit ici que pour ajouter 2 segments, il a fallu entrer 2 fois le point B.

### Modèle topologique

Le modèle topologique, par contre, décrit toutes les relations entre les segments et supprime la redondance d'information. Au détriment de l'efficacité de l'affichage, ces structures enregistrent les données de façon à accélérer l'exécution de requêtes spatiales.

L'image ci-dessous montre que chaque point a été stocké dans une table avec des informations le décrivant. Ensuite, une autre table s'occupe de lier ces divers points pour créer un ensemble complexe.



## Les étapes des systèmes d'information géographique

---

Un système d'information géographique se décompose en 5 étapes ou actions nommées les 5A : abstraction, acquisition, archivage, analyse et affichage.

### Abstraction

L'abstraction est en fait la modélisation du monde réel sous la forme d'un schéma conceptuel. C'est lors de cette étape, que l'on va découper le monde en couche et définir le modèle à utiliser. C'est sur ce schéma que la base de données sera créée et que l'outil adapté devra être défini.

### Acquisition

L'acquisition consiste en la saisie des données géographiques sous forme numérique. Ces informations peuvent être trouvées auprès de divers organismes. Si les informations recherchées n'existent pas, il est possible de les créer soit même grâce à diverses techniques :

- Scannage de documents existants
- Acquisition de photos, d'images satellites
- Acquisition à partir de données alphanumériques
- Acquisition sur le terrain (GPS,...)

### Archivage

Les données qui ont été acquises doivent être stockées et facilement accessibles. De plus, ces données doivent pouvoir être mise à jour régulièrement et de nouvelles données doivent pouvoir être ajoutée sans problème.

### Analyse

La raison d'être des SIG n'est pas seulement l'affichage de cartes mais également d'être un outil permettant d'analyser le monde pour donner un support de décision lors de choix scientifiques, économiques, politiques,...

Par exemple, il pourrait être intéressant lors de la construction d'une autoroute, de connaître le nombre d'habitations situées sur une zone de 100 mètres de part et d'autre de cette future autoroute. Cela permettrait d'envisager le tracé le moins dérangeant pour la population.

### Affichage

L'affichage cartographique n'est pas le cœur d'un SIG mais il permet d'avoir une représentation plus parlante d'une situation. L'utilisateur final aura ainsi accès aux informations spatiales de manière plus conviviale.

## Les domaines d'utilisation des SIG

---

Il y a quelques années la mise en place de SIG paraissait réservée aux grandes structures. Aujourd'hui les progrès informatiques et les possibilités offertes en matière de gestion et d'analyse, conduisent à la généralisation de l'outil à tous les échelons du territoire.

Ainsi, on trouve des SIG dans bons nombres de domaines :

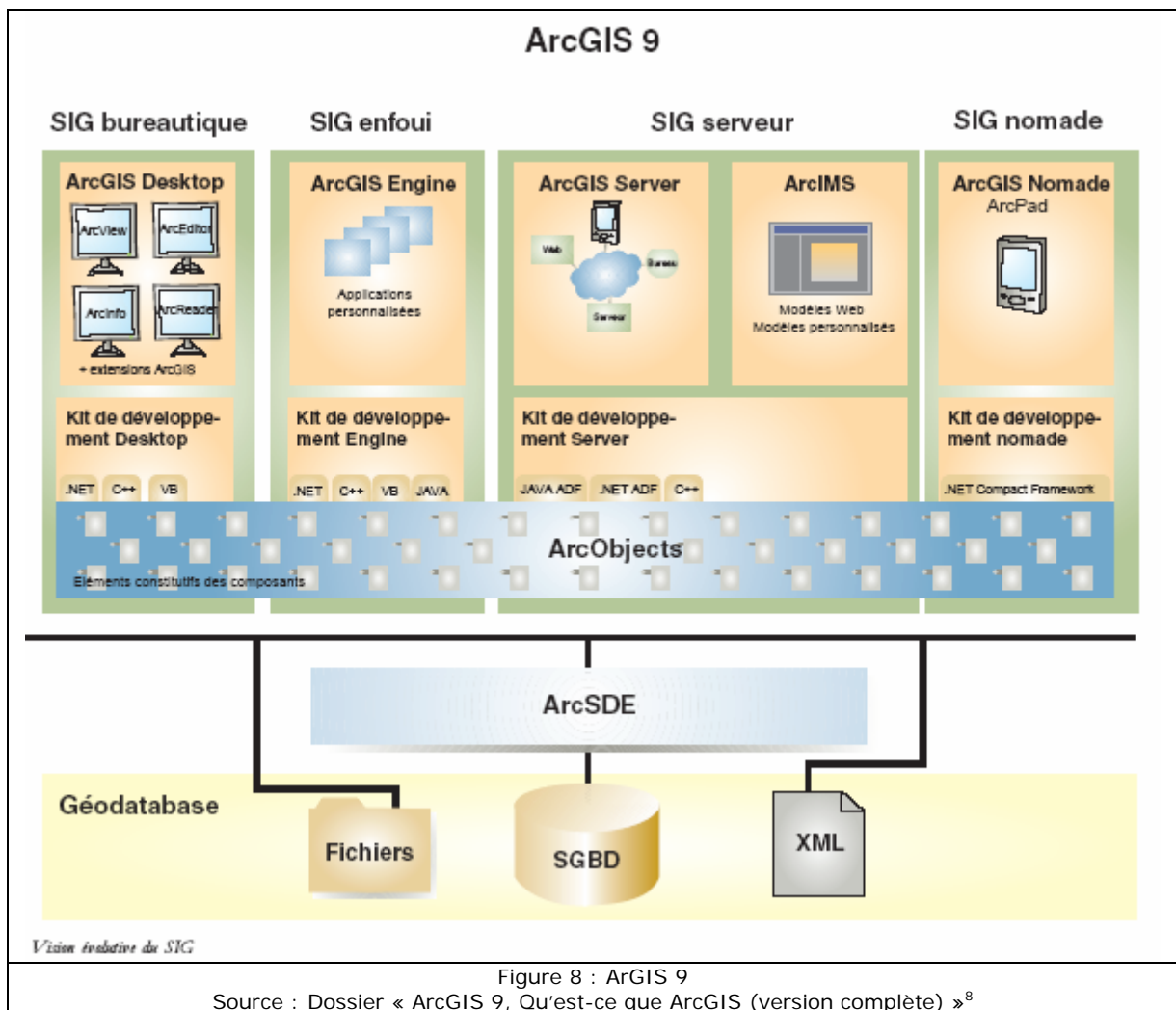
- aménagement du territoire et urbanisme
- agriculture
- industrie pétrolière,...
- prévention des risques naturels (inondations, feux de forêt, pollution,...)
- infrastructure routière
- construction publique
- gestion de l'eau
- ...

## Comparatif de quelques logiciels SIG du marché

### ArcGIS

ArcGIS est une suite logicielle complète créée par la société ESRI. Cette suite a la capacité de gérer un système d'information géographique dans son intégralité car elle s'occupe du géotraitement, de la géovisualisation et de la géodatabase.

La figure 8 ci-dessous montre les divers logiciels existants pour la mise en place d'un SIG.



<sup>8</sup> <http://www.esrifrance.fr/produit/arccgis.asp>

## Descriptif des outils ArcGIS

### *Bureautique*

ArcGIS Desktop est un outil de création et de gestion d'informations. Il peut être acheté sous forme de trois logiciels différents qui permettent d'effectuer des tâches comme la cartographie, l'analyse géographique, la gestion des données, la visualisation et le géotraitement :

- ArcView : Il offre des outils de cartographie interactive, d'utilisation des données et d'analyse complets, accompagnés de fonctions de mise à jour et de géotraitement.
- ArcEditor : Il permet l'administration et la mise à jour avancée des données géographiques dans les bases de données graphiques. Les fonctionnalités d'ArcView sont intégrées dans ce logiciel.
- ArcInfo : Cet outil regroupe l'ensemble des fonctions présentes dans ArcView et ArcEditor. Il intègre en plus, le géotraitement avancé et les anciennes applications ArcInfo Workstation.

### *Serveur*

ArcGIS propose trois produits pour serveur : ArcSDE, ArcIMS et ArcGIS Server.

- ArcSDE est un serveur de données spatiales qui permet à une application cliente (ArcIMS, ArcGIS Desktop,...) de stocker, gérer et utiliser les informations géographiques de nombreux systèmes de gestion de bases de données relationnelles grâce à une passerelle.
- ArcIMS est un serveur cartographique Internet évolutif permettant de publier des cartes, des données et des métadonnées SIG. Il propose un accès à de nombreux portails de catalogues.
- ArcGIS Server est un serveur d'application permettant d'élaborer des applications SIG côté serveur.

### *Moteur*

ArcGis Engine est une bibliothèque complète de composants SIG qui peuvent être incorporés dans des applications personnalisées. ArcGIS Engine fonctionne sous Windows, UNIX et Linux et prend en charge un certain nombre d'environnements de développement d'applications, comme Visual Basic 6, Microsoft Visual Studio .NET et de nombreux environnements de développement Java, y compris ECLIPSE et JBuilder.

### *Nomade*

ArcGIS propose le logiciel ArcPad pour le matériel Windows Mobile. Il est ainsi possible pour tout utilisateur d'accéder aux bases de données et aux fonctions de cartographie ainsi que d'intégrer les informations recueillies sur place grâce au système GPS à partir de périphériques portables et nomade.

### *Composants*

ArcGIS est basé sur une bibliothèque commune de composants logiciels SIG modulaires, appelés ArcObjects, se basant sur la technologie .COM. Ces composants peuvent être liés à d'autres logiciels ArcGIS pour permettre de créer des applications aux fonctions SIG complètes.



## MapInfo

---

MapInfo<sup>9</sup> propose également une suite logicielle pour du mapping, du routing ou du geocoding. Les principaux logiciels sont MapInfo Professional, MapBasic et MapXtreme.

### Descriptif des outils MapInfo

#### *MapInfo Professional*

MapInfo Professional est un logiciel permettant d'afficher et de manipuler tout type de données géographique ou alphanumériques pour en extraire des informations. Il permet de faire des analyses thématiques, des représentation de données, des accès à des bases de données distantes (Access, Oracle, SQL Server,...). C'est aussi une puissante interface permettant de faire des requêtes spatiales simples et avancées et d'échanger, d'intégrer et d'exporter des données.

#### *MapBasic*

MapBasic est un environnement de développement Basic, facilement intégrable dans des IDE comme Visual Studio. Il permet de créer des applications utilisables avec MapInfo Professional.

#### *MapXtreme*

MapXtreme est une composant de développement intégré pour la plate-forme .Net de Microsoft. Il permet de mettre en place des applications cartographiques performantes et robustes sur le web ou pour bureau.

---

<sup>9</sup> <http://www.mapinfo.com>

## Comparatif ArcView et MapInfo Professional

Pour permettre une meilleure vision entre les deux solutions, voici un comparatif entre les deux logiciels de représentation graphique :

<b>ArcView 9.1<sup>10</sup></b>	<b>MapInfo Professional 8.0<sup>11</sup></b>
1'500 \$	1'495 \$
512 MB Ram, 1GHZ	32 MB Ram, 103 MB espace disque
Windows 2000/XP	Windows 2003 Server/2000/XP
Doit posséder la partie ArcSDE pour se connecter à une base de données distante. Il ne peut que créer et modifier des données simples dans ces bases (points, lignes, polygones)	Accès ODBC direct aux principales bases de données distantes. Mise à jour des bases de données distantes avec gestion des conflits lors d'accès concurrents.
Cartes peuvent être sauveées, imprimées, exportées et intégrées dans des documents ou des applications.	Impression des cartes, mise en page de documents, visualisation que d'une partie d'une carte. Intégration dans tous logiciels client OLE grâce au serveur OLE
Raster/vecteur	Raster/vecteur
Rapport et chart en 3D	Editeur de rapports, graphique 3D, légende cartographique
Outils d'analyse prêt-à-l'emploi : ArcMap, ArcCatalog, ArcToolbox, ModelBuilder,...	Nombreuses fonctions de calcul, sectorisation, création de zone tampon, sélection d'objets, fusion et découpage de cartes,
Construction de modèles, scriptes et workflow	Création d'une bibliothèque de modèles, sauvegarde des modèles d'analyse thématique, sauvegarde des requêtes SQL,
Intégration de tous types de données (démographique, imagerie, WebServices, multimedia,...) et de plus de 70 différents formats.	Le Traducteur Universel vous permet de convertir tous types de fichiers cartographiques dans/depuis les principaux formats du marché : AutoCAD (DWG/DXF, y compris version 14 et attributs), ESRI (Shape, e00) et Intergraph MicroStation Design, DGN MSLinks et également les données issues du logiciel Atlas GIS.
Données disponibles <ul style="list-style-type: none"> <li>- Carte de base : Canada, Europe, Mexico et USA</li> <li>- Données commerciales de ESRI, DMTI Spatial,...</li> <li>- Données sur les routes américaines</li> <li>- ...</li> </ul>	Données disponibles <ul style="list-style-type: none"> <li>- Monde, Europe, France</li> <li>- Administration (Région de France)</li> <li>- Urbaines (Région de France)</li> <li>- Routière (Région de France)</li> <li>- Raster (Région de France)</li> <li>- Démographie (Région de France)</li> <li>- ...</li> </ul>
Pour avoir une démo, il faut demander le CD à ESRI	Démo 20 jours téléchargeable sur le site de MapInfo

<sup>10</sup> <http://www.esri.com/software/arcgis/arcview/about/features.html>

<sup>11</sup> MapInfo Professional  
<http://www.i2g.fr/pdf/mapinfo.pdf>

## Oracle Application Server MapViewer

---

MapViewer<sup>12</sup> est un service J2EE qui permet l'affichage de cartes en utilisant des données géo-spatiales managées par Oracle Spatial ou Oracle Locator. MapViewer propose des services et des outils qui cachent la complexité des requêtes et de la cartographie. De plus, il apporte des options pouvant s'adapter aux besoins des clients.

MapViewer possède un moteur (bibliothèque de classe Java) ainsi qu'une API XML qui lui fournit une interface programmable. L'application se connecte grâce aux classes Java à la base de données Oracle par JDBC. Ensuite, les métadonnées des cartes sont chargées depuis la base et sont appliquées aux données spatiales.

Fonctionnalités :

- Support d'Oracle Spatial GeoRaster
- Support du modèle de données réseau d'Oracle Spatial
- Support du modèle de données topologique d'Oracle Spatial
- Possibilité de mettre des versions aux tables de la base de données. Ainsi, on peut faire la requête de la carte par rapport à différentes versions des données.
- Possibilité d'avoir des sources de données multiples pour l'analyse d'une carte
- Cartes au format SVG qui permet lors du déplacement du curseur sur la carte, d'afficher des informations relatives au lieu désiré en pop-up
- Support le protocole WMS (Web Map Service) d'OpenGIS
- Supporte les cartes au format JPG et PNG

## QGIS (Quantum GIS)

---

QGIS<sup>13</sup> est un système d'information géographique open source fonctionnant sur Linux, Mac OSX et Windows, avec cependant des fonctionnalités plus matures sur Linux que sur Windows où quelques problèmes de stabilités subsistent toujours. Le but initial de QGIS était de permettre la visualisation de données spatiales, mais très vite de nouvelles fonctionnalités ont été ajoutées. Quantum GIS a été développé sous la GNU Public License (GPL), ce qui signifie que les utilisateurs peuvent avoir accès et modifier en toute liberté le code source et qu'ils ont la garantie que le logiciel sera toujours gratuit.

QGIS possède beaucoup de fonctionnalités de ce genre de logiciel. Voici la liste des fonctionnalités principales :

- Supporte la base de données PostgreSQL grâce à son extension spatiale OpenGIS ainsi qu'accès ODBC
- Supporte les fichiers shapfiles de ESRI ainsi que d'autres formats de vecteur, incluant les fichiers MapInfo
- Affichages des couches vecteurs
- Possibilité de faire des compositions de cartes
- Supporte le format raster. Possibilité d'afficher les données raster comme des modèles digitaux d'élévations, des photographies aériennes ou des imageries de terrain
- Possibilité de modifier, de créer et d'ajouter des métadonnées aux couches vectorielles
- Possibilité de créer et d'ouvrir des projets
- Affichage des attributs des tables

---

<sup>12</sup> Oracle Application Server – MapViewer user's guide 10g release 2  
[http://www.oracle.com/technology/products/mapviewer/mapviewer\\_doc\\_index.html](http://www.oracle.com/technology/products/mapviewer/mapviewer_doc_index.html)

<sup>13</sup> Quantum GIS user guide  
[http://qgis.org/index.php?option=com\\_content&task=view&id=106&Itemid=65](http://qgis.org/index.php?option=com_content&task=view&id=106&Itemid=65)

## SharpMap

---

SharpMap<sup>14</sup> est un interpréteur de carte open source utilisable dans des applications web ou Windows. Tout comme QGIS, SharpMap a été publié sous GNU Public License.

Fonctionnalités :

- Peut être intégré à toutes applications utilisant un langage .Net, comme C#, VB.NET et J#
- Supporte la base de données PostreSQL/PostGIS, MS SQL Server, Oracle ainsi que les accès OleDb
- Supporte les fichiers shapefiles d'ESRI, les fichiers MapInfo,...
- Supporte les images raster ECW et JPEG2000 ainsi que d'autres formats grâce à un plugin
- Supporte les types de données définies par la norme Simple Features Specifications d'OpenGIS
- Gestion de labels
- Cartes thématiques avancées
- Facilement extensible à des fournisseurs de données, des types de couches et des types de géométries additionnels

## Récapitulatif

---

	ArcView	MapInfo Professional	QGIS	MapViewer	SharpMap
Oracle Spatial	Oui	Oui	Non	Oui	Oui
Oracle Locator	Oui	Oui	Non	Oui	Oui
SQL Server	Oui	Oui	Non	Non	Oui
MySQL	Non	Non	Non	Non	Non
PostGIS	Oui/non *	Non	Oui	Non	Oui
	1'500 \$	1'495 \$	OpenSource	Gratuit	OpenSource

\* l'extension DataInteroperability d'ArcGIS permet de visualiser les données PostGIS mais pas de les éditer. ESRI est en court de portage d'ArcSDE sous PostreSQL

---

<sup>14</sup> <http://www.codeplex.com/Wiki/View.aspx?ProjectName=SharpMap>

## Open Geospatial Consortium (OGC)

Open Geospatial Consortium est une organisation internationale à but non lucratif, créée en 1994, dédiée au développement de systèmes ouverts en géomatique<sup>15</sup>. Plus de 300 compagnies, organismes gouvernementaux ou universités du monde entier participent à l'élaboration de spécifications OpenGIS (normes et standards) permettant l'accès à des données spatiales complexes par n'importe quel type d'applications. Ces spécifications sont nécessaires à l'interopérabilité des systèmes d'information géographique (SIG). Cet organisme était autrefois connu sous le nom de Open GIS Consortium.

Liste de quelques membres :

- ESRI
- Oracle
- Autodesk
- Google
- MapInfo Corporation
- MIT
- EPFL
- NASA
- ...

Actuellement, plus d'une douzaine de spécifications ont été approuvées. Les recommandations les plus importantes faites par l'OGC sont :

- WMS : Web Map Service
  - o Il décrit le comportement d'un service web dont le but est l'affichage dynamique de cartes géoréférencées à partir de données spatiales. WMS fournit les cartes sous forme d'image mais ne permet pas la récupération des données géographiques elles-mêmes. Les fonctionnalités principales sont la récupération côté client des cartes, la description au client des cartes fournies par le service et l'envoi de requêtes sur les objets géographiques de ces cartes.
- WFS : Web Feature Service
  - o Il définit les interfaces d'un service web permettant d'extraire et de mettre à jour des données spatiales encodées en GML (Geography Markup Language).
- WCS : Web Coverage Service
  - o Extension de l'interface WMS permettant l'accès à des données raster (images satellites, MNT<sup>16</sup>,...)
- CS-W : Catalog Service Web
  - o Il définit une interface commune qui permet à diverses applications de faire des opérations de recherches, des requêtes,... sur des serveurs de catalogues.

---

<sup>15</sup> La géomatique regroupe l'ensemble des disciplines et moyens informatiques permettant de représenter, d'analyser et d'intégrer des données géographiques. Elle consiste donc en au moins trois activités distinctes : la collecte, le traitement et la diffusion des données.

<sup>16</sup> Un modèle numérique de terrain (MNT) est une représentation de la topographie d'une zone terrestre (rendu tridimensionnel) sous une forme adaptée à son utilisation par un ordinateur.

- SFS : Simple Features SQL
  - o Il décrit une interface qui permet un accès transparent à des données géographiques. Il spécifie la publication, le stockage et l'accès aux données spatiales (Point, LineString, Polygon, ...) ainsi que les fonctions permettant la gestion de ces données.
- GML : Geography Markup Language
  - o C'est un langage dérivé du XML permettant de décrire des objets géographiques, des systèmes de projection, ainsi que la géométrie, la topologie, le temps, les unités de mesures et les attributs des objets géographiques. Ce profil se veut un moyen d'assurer l'interopérabilité entre applications.

## La norme OpenGIS : Simple Features Specifications for SQL

La norme Simple Features Specifications for SQL a été créée pour définir un schéma standard SQL qui supporterait le stockage, la récupération, les requêtes et les mises à jour des collections de données spatiales via l'API ODBC.

Cette norme standardise :

- le nom et la définition des géométries
- le nom, la signature et la définition des fonctions liées aux géométries

Cette partie du rapport repose exclusivement sur le document « OpenGIS Simple Features Specifications for SQL<sup>17</sup> » créé par l'Open Geospatial Consortium.

### Approche

---

Le stockage des données spatiales dans une base de données relationnelle se fait par l'intermédiaire d'une table appelée « feature table » contenant au moins une colonne de type géométrique.

Pour l'implémentation de ces tables, deux environnements sont proposés par la norme SFS : SQL92 et SQL92 avec types géométriques.

L'environnement SQL92 définit le schéma pour le stockage des données spatiales sans l'utilisation de nouveaux types de données.

L'environnement SQL92 avec types géométriques est une extension de l'environnement SQL92 où les données spatiales sont gérées par un nouveau type de données, la classe Geometry. La norme SFS a défini un modèle ainsi que les fonctions relatives à ces géométries.

Pour être compatible avec la norme Simple Features Specification for SQL, une base de données relationnelle doit implémenter une des trois alternatives suivantes (1a,1b ou 2):

1° Implémentation de « feature tables » avec un environnement SQL92

- a) en utilisant des types numériques pour le stockage des données spatiales et l'accès ODBC
- b) en utilisant des types binaires pour le stockage des données spatiales et l'accès ODBC

2° Implémentation de « feature tables » avec un environnement SQL92 avec type géométrique supportant un accès ODBC binaire et textuel

---

<sup>17</sup> OpenGis Simple Features Specification for SQL Revision 1.1  
<http://www.opengeospatial.org/standards/sfs>

## Modèle Object Geometry

Les types de géométrie définis par OpenGIS sont décrits dans l'image ci-dessous. Les différentes géométries sont classées hiérarchiquement par rapport à la classe Geometry, qui est la classe de base (root). Certaines de ces classes ne sont pas instanciables.

Chaque géométrie est associée à un système de référence spatiale, qui décrit l'espace de coordonnées dans lequel est défini l'objet géométrique, comme la latitude/longitude ou la zone UTM.

Les géométries sont définies dans cette norme comme étant restreintes à des objets de 0, 1 ou 2 dimensions. Le point est de dimension 0, la courbe et ses sous-classes de dimension 1 et la surface et ses sous-classes de dimension 2. La classe GeometryCollection possède la dimension des géométries qui la compose.

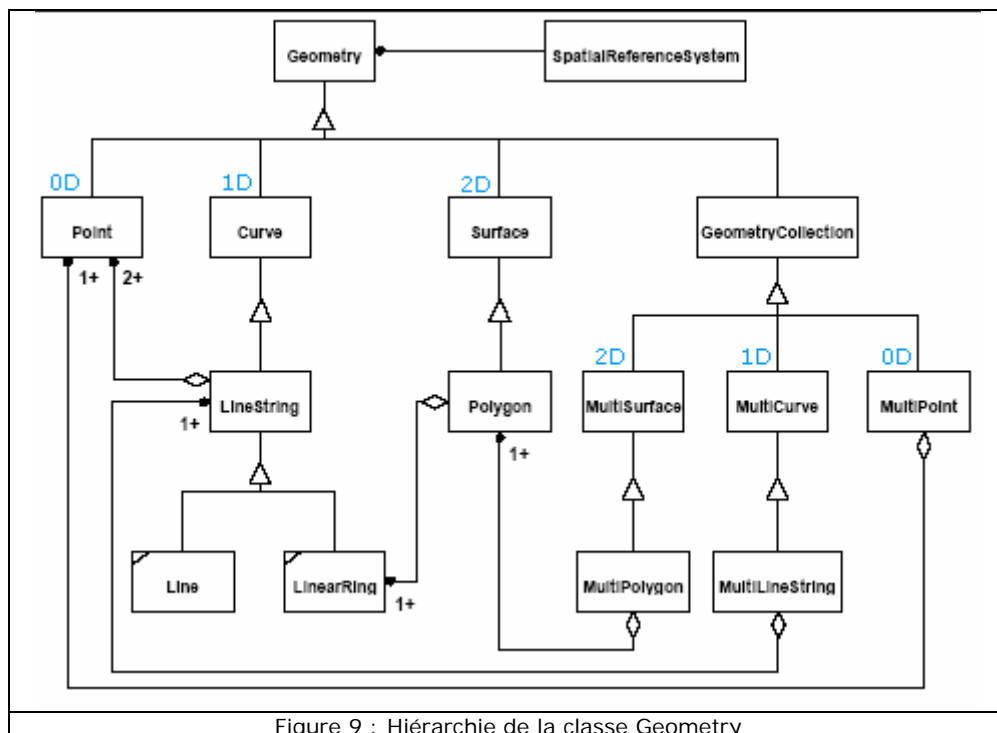


Figure 9 : Hiérarchie de la classe Geometry

## Geometry

Geometry est la classe racine de la hiérarchie. Elle possède une série de propriétés qui sont communes au reste des géométries :

- type : chaque géométrie appartient à une classe de géométrie instanciable (Point, LineString,...)
- SRID : identifiant du système de référence spatial
- coordonnée dans le système de référence spatial : représentée par des doubles. Tout objet non vide contient au moins une coordonnée x et une coordonnée y
- intérieur (espace utilisé par la géométrie), extérieur (espace non occupé par la géométrie) et frontière (interface entre l'intérieur et l'extérieur)
- enveloppe : la forme géométrique minimum qui entoure une géométrie
- simple ou non : si ne passe pas 2 fois par le même point (pour les LineString, MultiPoint, MultiLineString)
- fermé ou non : si le point d'origine est le point d'arrivée (pour LineString et MultiLineString)



- vide ou non : une géométrie vide n'a pas de point. Son intérieur, extérieur et sa frontière ne sont pas définis.
- dimension : -1 (vide), 0, 1 ou 2

Toutes les classes héritent de Geometry cependant, celle-ci n'est pas instanciable.

## GeometryCollection

C'est une géométrie qui est une collection d'une ou plusieurs géométries. Tous les éléments dans une GeometryCollection doivent appartenir au même système de référence spatiale.

## Point

Un point est une géométrie à 0 dimension qui représente un lieu unique dans l'espace de coordonnées. Un point a une coordonnée x et une coordonnée y.

## MultiPoint

Collection de points non connectés et non ordonnés

## Curve

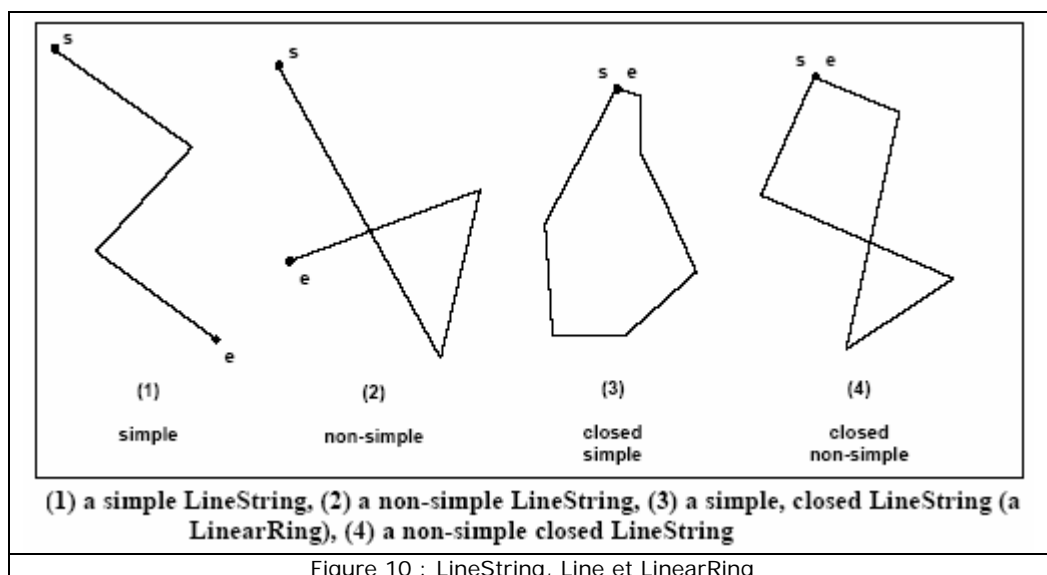
Une courbe est une géométrie à 1 dimension stockée en tant que séquence de points. Elle diffère de la LineString par le fait que sa représentation n'est pas linéaire.

## LineString, Line, LinearRing

Une LineString est une courbe avec une interpolation linéaire entre les points.

Une Line est une LineString avec exactement 2 points.

Une LinearRing est une LineString qui est fermée et simple.

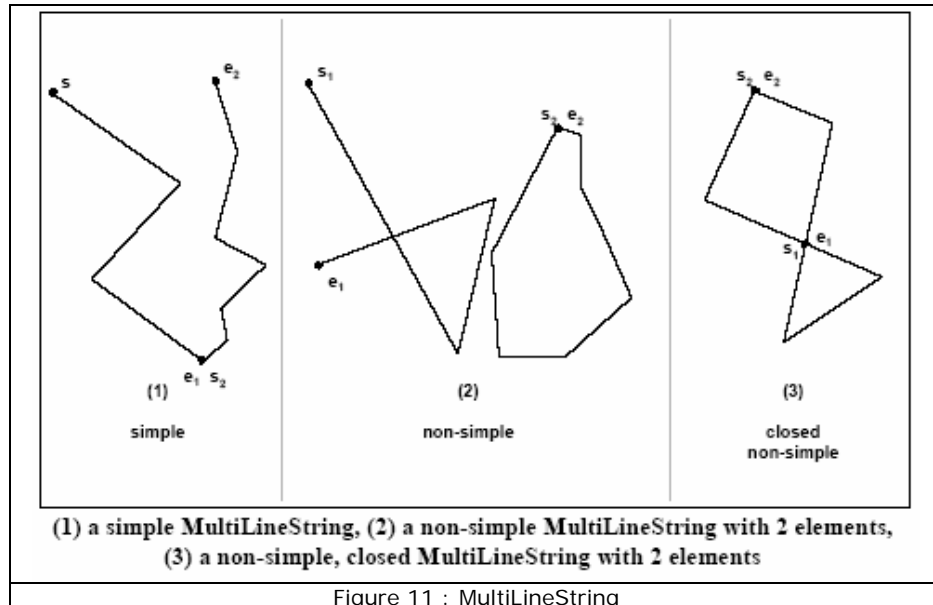


## MultiCurve

Collection de courbes. Cette classe n'est pas instanciable mais définit une série de fonctions pour les classes descendantes.

## MultiLineString

C'est une MultiCurve dont les éléments sont des LineString



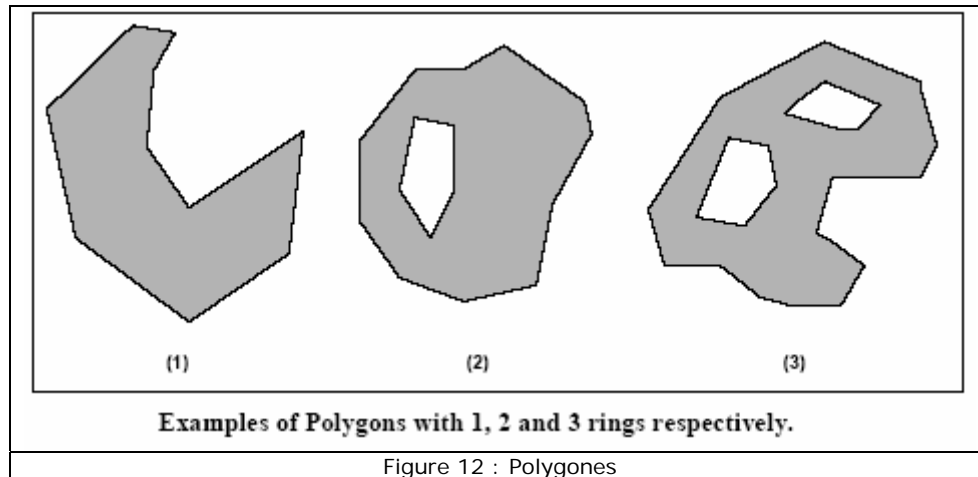
## Surface

Une surface est un objet géométrique à 2 dimensions. Elle est constituée d'une frontière extérieure et de 0 ou plusieurs frontières intérieures.

Les surfaces à 3 dimensions sont formées en « cousant » ensemble des surfaces simples le long de leurs frontières.

## Polygon

Un polygone est une surface plane. Si le polygone possède des frontières intérieures, elles seront définies comme des trous dans le polygone. Chaque trou ajoute un anneau au polygone.



## MultiSurface

Collection de surfaces qui ne doivent pas se croiser. Cette classe n'est pas instanciable, mais ses sous-classes le peuvent.

## MultiPolygon

MultiSurface dont les éléments sont des polygones.

## Relations entre les géométries

---

La norme SFS propose des méthodes booléennes permettant de tester les relations topologiques entre deux géométries spécifiques.

- Equals() : si l'enveloppe des géométries est la même
- Disjoint() : retourne vrai si les géométries n'ont pas d'intersection
- Intersects() : retourne vrai si les géométries ont une intersection (inverse de disjoint)
- Touches() : si une géométrie touche une autre
- Crosses() : si une géométrie traverse, rencontre une autre
- Within() : si une géométrie est à l'intérieur d'une autre
- Contains() : si une géométrie contient complètement une autre (inverse de within)
- Overlaps() : si une géométrie recouvre une partie d'une autre
- Relate() : si une géométrie est reliée à une autre par son intérieur, son extérieur ou sa frontière

## Fonctionnalités

---

Les classes enfant héritent des fonctionnalités de leur parent

### Fonction de base et de constructeur de géométrie

- GeomFromText : Création d'une géométrie à partir de sa représentation textuelle
- PointFromText : Création d'un point à partir de sa représentation textuelle
- ... : Chaque géométrie a sa fonction de création par représentation textuelle
- GeomFromWKB : Création d'une géométrie par sa représentation binaire
- PointFromWKB : Création d'une géométrie par sa représentation binaire
- ... : Chaque géométrie a sa fonction de création par représentation binaire
- AsText : retourne la représentation textuelle de la géométrie
- AsBinary : retourne la représentation binaire de la géométrie
- Dimension : retourne le nombre de dimension de l'objet
- GeometryType : retourne le type de la géométrie
- SRID : retourne un entier indiquant l'identifiant du système de coordonnées
- IsEmpty : si la géométrie est vide ou non
- IsSimple : la géométrie est simple si elle n'a aucune anomalie géométrique, comme par exemple une auto-intersection ou une auto-tangence
- Boundary : retourne la forme qui représente la frontière de la géométrie
- Envelope : retourne l'enveloppe (rectangle) minimum d'une géométrie

### Point

- X : retourne l'abscisse du point
- Y : retourne l'ordonnée du point

### Curve

- StartPoint : retourne le premier point de la courbe
- EndPoint : retourne le dernier point de la courbe
- IsClosed : retourne oui si le StartPoint = EndPoint
- IsRing : retourne oui si la géométrie est fermée et qu'elle est simple (sans anomalie)
- Length : retourne la longueur de la courbe

### LineString

- NumPoints : retourne le nombre de points contenus dans la ligne
- pointN : retourne les coordonnées du point à la position N

### Surface

- centroid : retourne le centre mathématique de l'objet
- PointOnSurface : retourne un point qui est dans l'objet
- Area : retourne l'air de l'objet

### Polygon

- ExteriorRing : retourne l'anneau extérieur du polygone
- NumInteriorRing : retourne le nombre d'anneau intérieur du polygone
- InteriorRingN : retourne le n-ième polygone

### GeomCollection

- NumGeometries : retourne le nombre de géométrie présentes dans la collection
- GeometryN : retourne la n-ième géométrie

### Multicurve

- IsClosed : retourne oui si pour toutes les courbes de la collection, le StartPoint = EndPoint
- Length : retourne la somme des longueurs des géométries contenues dans la collection

### MultiSurface

- Centroid : retourne le centre mathématique de la MultiSurface
- PointOnSurface : retourne un point contenu dans la MultiSurface
- Area : retourne l'air de la MultiSurface

### Opérateurs spatiaux

- Distance : retourne la distance entre 2 géométries
- Intersection : retourne la géométrie qui est l'intersection de 2 autres
- Difference : Ensemble des points représentant la différence de 2 formes
- Union : retourne la géométrie qui est l'union de 2 autres
- SymDifference : géométrie étant la différence géométrique de 2 géométries
- Buffer : retourne la géométrie dont la distance à la forme est inférieure ou égale à une distance choisie
- ConvexHull : retourne l'enveloppe convexe de la forme géométrique

## La représentation binaire des géométries (WKB)

---

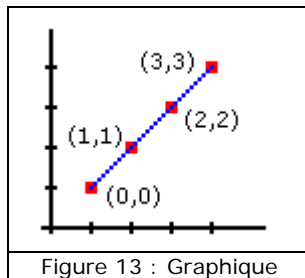
Représente une géométrie comme un flux de bytes et permet de les échanger à travers un client ODBC et une base de données SQL.

Ce flux est obtenu en sérialisant une instance géométrique en une séquence de type numérique (integer ou double) puis en sérialisant chaque type numérique en une séquence de byte utilisant un encodage de type numérique soit NDR soit XDR.

Une représentation binaire est construite ainsi :

- format d'encodage (XDR = 0, NDR = 1)
- type de géométrie
  - o Point = 1
  - o LineString = 2
  - o Polygon = 3
  - o MultiPoint = 4
  - o MultiLineString = 5
  - o MultiPolygon = 6
  - o GeometryCollection = 7
- la géométrie
  - o Point
    - X
    - Y
  - o LineString
    - Nombre de point
    - Points
  - o Polygon
    - Nombre d'anneau
    - Nombre de points (par anneau)
    - Points (par anneau)
  - o MultiPoint
    - Nombres de points
    - Points
  - o MultiLineString
    - Nombre de LineString
    - Nombre de points (pour chaque LineString)
    - Points
  - o MultiPolygon
    - Nombre de polygone
    - Nombre d'anneau (par polygone)
    - Nombre de points (pour chaque anneau)
    - Points
  - o GeometryCollection
    - Nombre de géométrie
    - Description de chaque géométrie selon informations ci-dessus

En réalité, chaque géométrie est basée sur le point qui est constitué de deux « double ». Ensuite, toute géométrie est construite en utilisant des représentations de géométries déjà définies.



0 | 2 | 4 | 0 0 | 1 1 | 2 2 | 3 3 (représentation décimale qu'il faudra transformer en binaire)

Encodage XDR

Type LineString

Comprenant 4 points (0,0) (1,1) (2,2) (3,3)

## La représentation textuelle des géométries

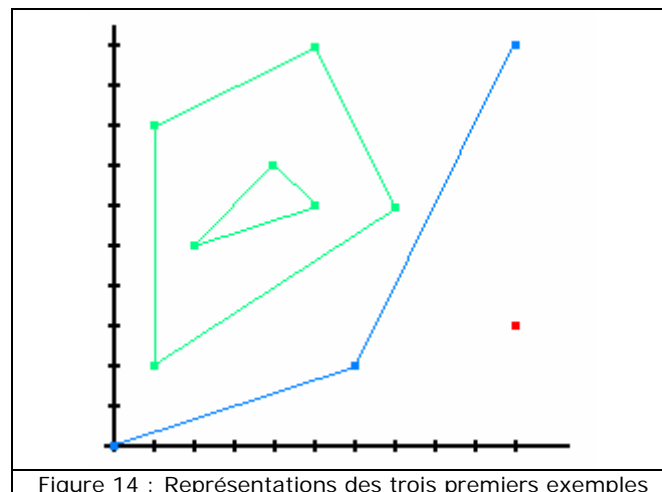
La représentation WKT permet d'échanger des géométries grâce au format ASCII.

Elle est structurée ainsi :

- Nom du type de géométrie
- Les coordonnées des points formant cette géométrie entre parenthèse
- Pour signifier qu'une géométrie est fermée, on lui indique comme dernier point le même que celui de départ

Voici quelques exemples :

- Point(10 3) (rouge)
- LineString(0 0, 6 2, 10 10) (bleu)
- Polygon((1 2, 1 8, 5 10, 7 6, 1 2),(2 5, 4 7, 5 6, 2 5)) (vert)



- MultiPoint(10 3, 4 5)
- MultiLineString((0 0, 10 10, 20 15),(25 20, 30 35, 40 40))
- GeometryCollection(Point(10 10), Point(15 10), LineString(0 0, 10 10, 20 15))

## La représentation textuelle des systèmes de référence spatiaux (WKT)

---

Apporte un standard textuel pour la représentation des systèmes de références spatiales. Un système de référence spatiale peut être un système de coordonnées géographiques (latitude/longitude), projeté (x, y) ou géocentrique (x, y, z).

Selon le système de coordonnées choisi, on utilisera un mot clé différent : PROJCS pour un système projeté, GEOGCS pour un système géographique et GEOCCS pour un système géocentrique. Ensuite, le système sera défini par un nom et d'autres informations comme l'unité angulaire de mesure, le premier méridien, le facteur de conversion,...

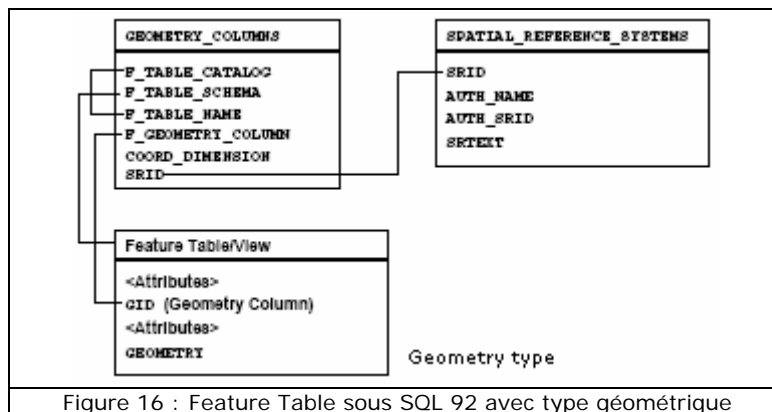
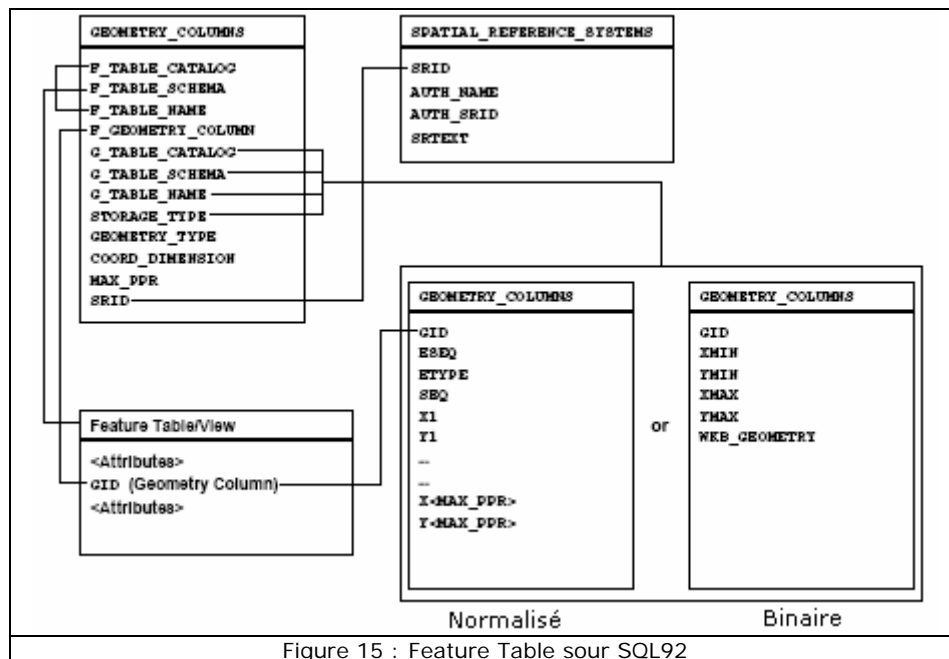
Exemple :

```
PROJCS['NAD_1983_UTM_Zone_10N', --nom du système de coordonnées
    GEOGCS['GCS_North_American_1983', --nom du système de coordonnées
        géographiques
        DATUM['D_North_American_1983', --nom des données
            SPHEROID['GRS_1980',6378137,298.257222101]],
        PRIMEM['Greenwich',0], -- le premier méridien
        UNIT['Degree',0.0174532925199433]], -- unité de mesure angulaire
    PROJECTION['Transverse_Mercator'], --projection sur la carte (ici UTM)
    --liste de paramètres
    PARAMETER['False_Easting',500000.0],
    PARAMETER['False_Northing',0.0],
    PARAMETER['Central_Meridian',-123.0],
    PARAMETER['Scale_Factor',0.9996],
    PARAMETER['Latitude_of_Origin',0.0],
    UNIT['Meter',1.0]] --unité de mesure linéaire
```



## Implémentation des « feature tables »

Les informations concernant les données spatiales sont stockées dans une « feature table ». Cependant, la mise en place d'une telle table varie selon l'environnement choisi. Ci-dessous se trouvent deux schémas de base de données nécessaires pour supporter la norme SFS. Le premier schéma est celui de l'environnement SQL92 et le deuxième celui de l'environnement SQL92 avec types géométriques.



## Description des schémas

---

### Feature table

Dans l'environnement SQL92, une « feature table » est une table ayant une ou plusieurs clés étrangères référencées sur une table géométrique (Geometry\_Columns).

Dans un environnement SQL92 avec types géométriques, une « feature table » est une table ayant une ou plusieurs colonnes dont le type de données est une des géométries définies par OpenGIS.

Exemple de « feature tables » dans chaque environnement :

#### Environnement SQL92

<i><b>Feature table</b></i>	<u>Ville</u>	GID (FK)
	String(25)	Integer
	Sion	1
	Sierre	2

<i><b>Geometry_Columns</b></i>	<u>GID</u>	Xmin	Ymin	Xmax	Ymax	WKB_Geometry
	Integer	Double	Double	Double	Double	Binary
	1	1	1	1	1	0010100010...
	2	2	3	2	3	0100010100...

Cette table sera présentée plus en détail dans le chapitre xx

#### Environnement SQL92 avec types géométriques

<i><b>Feature table</b></i>	<u>Ville</u>	Lieu
	String(25)	Point
	Sion	Point(1,1)
	Sierre	Point(2,3)

## Table Geometry\_Columns (métadonnées)

Chaque colonne géométrique doit être représentée dans la table de métadonnées Geometry\_columns. Elle permet de garder les informations suivantes :

F\_Table\_Catalog, F\_Table\_Schema, F\_Table\_Name : nom, catalogue et schema de la « feature table »

F\_Geometry\_Column : le nom de la colonne dans la « feature table » qui est la colonne géométrique. (En SQL92, cette colonne sera une référence clé étrangère dans la table géométrique)

G\_Table\_Catalog, G\_Table\_Schema, G\_Table\_Name : le nom, le catalogue et le schéma de la table géométrique.

Storage\_type : le type de stockage utilisé pour cette colonne : numérique ou binaire

Geometry\_Type : le type de géométrie de la colonne :

- 0 = geometry
- 1 = Point
- 2 = Curve
- 3 = LineString
- 4 = Surface
- 5 = Polygon
- 6 = Collection
- 7 = MultiPoint
- 8 = MultiCurve
- 9 = MultiLineString
- 10 = MultiSurface
- 11 = MultiPolygon

Coord\_Dimension : le nombre de dimensions utilisées par la géométrie

Max\_Ppr : le nombre de points stockés en tant que colonne de coordonnées dans la « feature table » (ce champ est utile seulement lors de l'emploi de la version de représentation normalisée des données. Il permet de définir le nombre de colonnes utilisées pour la représentation des points d'une géométrie. Dans l'exemple ci-dessous, il est défini que 5 points serait décrit par lignes.)

GID	ESEQ	ETYPE	SEQ	X0	Y0	X1	Y1	X2	Y2	X3	Y3	X4	Y4
2	2	3	1	40	5	40	20	45	20	45	15	50	15
2	2	3	2	50	15	50	5	40	5	Nil	Nil	Nil	Nil

Figure 17 : Exemple de Feature Table

SRID : l'identifiant du système de référence spatiale. C'est une clé étrangère référencée dans la table Spatial\_ref\_sys

Dans l'environnement SQL92 avec types géométriques, toutes les colonnes présentées ci-dessus ne sont pas nécessaires. Seules les colonnes F\_Table\_Catalog, F\_Table\_Schema, F\_Table\_Name, F\_Geometry\_Column, Coord\_Dimension et SRID sont utilisées.

## Table Spatial\_Reference\_System

Chaque colonne géométrique est associée à un système de référence spatiale. Le système de référence spatiale identifie le système de coordonnées pour toutes les géométries de la colonne et donne une signification aux coordonnées numériques pour toutes instance de géométrie. Ex : latitude/longitude ou UTM Zone 10

La table Spatial\_reference\_system stocke des informations sur tous les systèmes des références spatiales de la base de données.

On trouve les champs suivants :

SRID = identifiant du système de référence spatiale (clé primaire)

AUTH\_NAME = nom de l'organisation qui gère ce système de référence spatiale (EPSG)

AUTH\_SRID = identifiant du système de référence spatiale défini par l'AUTH\_NAME

SRTEXT : description WKT (well-known text description) du système de référence spatiale (permet l'interopérabilité entre les clients)

## Tables contenant les géométries (SQL92 seulement)

Il y a deux implémentation pour stocker les informations géographiques avec l'environnement SQL92 : en utilisant le schéma normalisé ou le schéma binaire (WKB).

### *Normalisé*

Le schéma normalisé utilise un nombre fixe de colonnes pour le stockage de données spatiales. Chaque élément de la géométrie est défini par son identifiant unique de la géométrie (GID), sa séquence (SEQ), son type (ETYPE) et sa position dans la séquence (ESEQ)

Ainsi, une table représentant des géométries en format normalisé possèdera les champs suivants :

- GID : identifiant de la géométrie
- ESEQ : numéro de l'élément dans la géométrie
- ETYPE : type de la géométrie (1 = point, 2 = LineString, 3 = Polygon)
- SEQ : une géométrie peut être construite de plusieurs parties. Chacune est numérotée.
- X1 : première ordonnée
- Y1 : deuxième ordonnée
- ...
- X<max\_ppr> : xième ordonnée
- Y<max\_ppr> : yième ordonnée
- Attribut : d'autres attributs peuvent être ajoutés à cette table

Les coordonnées non utilisées doivent être mises à Null, ce qui permet de savoir qu'on est à la fin de la géométrie.

Si des géométries utilisent plus d'une ligne, alors le dernier point de la première ligne doit être le même que le premier point de la deuxième ligne.

### *Binaire*

Pour l'implémentation binaire, les informations importantes sont l'identifiant de la géométrie, le rectangle minimal entourant la géométrie, la représentation binaire de la géométrie ainsi que des attributs supplémentaires standard.

L'implémentation binaire de cette table est la suivante :

- GID : identifiant de la géométrie
- XMIN : la coordonnée x minimum de l'enveloppe de la géométrie
- YMIN : la coordonnée y minimum de l'enveloppe de la géométrie
- XMAX : la coordonnée x maximum de l'enveloppe de la géométrie
- YMAX : la coordonnée y maximum de l'enveloppe de la géométrie
- WKB\_Geometry : la représentation binaire de la géométrie
- Attributs : d'autres attributs peuvent être ajoutés à la géométrie.

*Exemple de représentation normalisée et binaire*

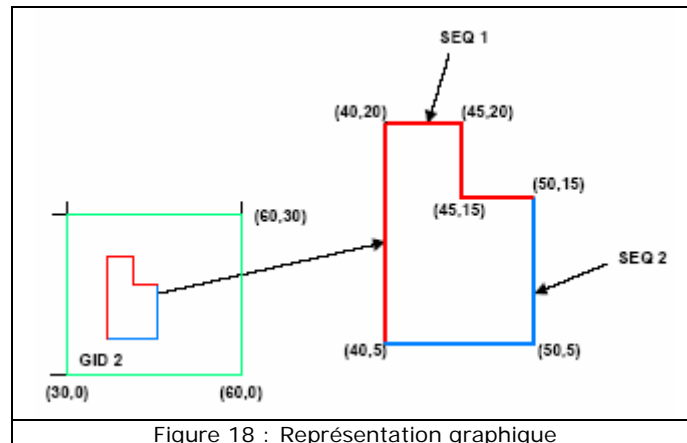


Figure 18 : Représentation graphique

GID	ESEQ	ETYPE	SEQ	X0	Y0	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	1	3	1	0	0	0	30	30	30	30	0	0	0
1	2	3	1	10	10	10	20	20	20	20	10	10	10
2	1	3	1	30	0	30	30	60	30	60	0	30	0
2	2	3	1	40	5	40	20	45	20	45	15	50	15
2	2	3	2	50	15	50	5	40	5	Nil	Nil	Nil	Nil
3	1	3	1	0	30	0	60	30	60	30	30	0	30
4	1	3	1	30	30	30	60	60	60	60	30	30	30

Figure 19 : Représentation normalisée

Exemple de la géométrie ci-dessus composée de 2 anneaux pour une représentation normalisée :

Cette géométrie a le numéro 2 → GID = 2

Elle est composée de 2 anneaux → ESEQ = 1 pour l'anneau extérieur et 2 pour l'anneau intérieur

Les deux anneaux sont de types polygone → ETYPE = 3

L'anneau extérieur peut être défini dans une ligne → SEQ = 1

L'anneau intérieur doit être divisé en 2 parties → 1<sup>er</sup> partie SEQ = 1 et 2<sup>e</sup> partie SEQ = 2

Pour l'anneau interne, il est donc nécessaire d'utiliser 2 lignes. Comme déjà mentionné plus haut, le dernier point de la ligne 1 doit être le même que le premier point de la ligne 2 (violet). De plus, les colonnes vides doivent être complétées par « Null » (jaune)

La même géométrie représentée en format binaire :

GID	XMIN	YMIN	XMAX	YMAX	GEOMETRY
1	0	0	30	30	< WKBGeometry >
2	30	0	60	30	< WKBGeometry >
3	0	30	30	60	< WKBGeometry >
4	30	30	60	60	< WKBGeometry >

Figure 20 : Représentation binaire

Le GID est le même que pour la représentation normalisée → GID 2

On va définir le rectangle (enveloppe) minimal qui entoure la géométrie. Dans cet exemple, le rectangle minimal correspond à l'anneau extérieur de la géométrie. On définit le rectangle en donnant ses coordonnées (xmin,ymin) et (xmax,ymax). → (30,0) et (60,30)

Finalement, la dernière colonne donne la représentation binaire de la géométrie. (cf. voir WKB)

0 3 2 5 0 30 30 30 60 30 60 0 0 30 7 50 5 40 20 45 20 45 15 50 15 50 5 40 5

Les chiffres présentés ici doivent bien sur être représentés de façon binaire. Cette représentation permet juste d'avoir une vision plus claire d'une géométrie.

# Utilisation d'Oracle Spatial

## Oracle Spatial

---

### Systèmes de coordonnées<sup>18</sup>

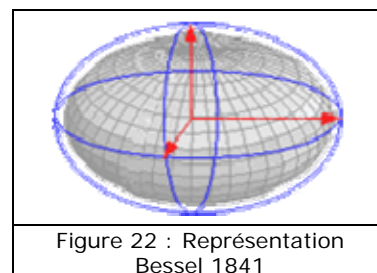
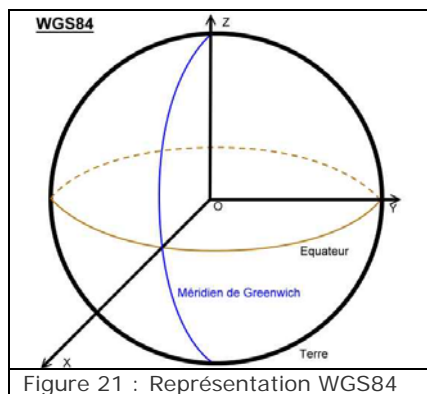
Un système de coordonnées ou système de références spatiales permet l'interprétation d'une série de coordonnées en tant que représentation d'une position (lieu) dans le vrai espace mondial.

Toute donnée spatiale a un système de coordonnées associé. Ce système peut être géoréférence (lié à une représentation spécifique de la terre) ou non géoréférencé (données cartésiennes, non liées à une représentation spécifique de la terre). Il est possible de transformer les données d'un système à un autre.

Il existe deux grandes familles de systèmes de coordonnées : les globaux et les locaux.

Les systèmes de référence terrestres globaux sont des systèmes de coordonnées cartésiennes (x,y,z) géocentriques dont l'origine se trouve au centre de gravité des masses terrestres M. L'axe Z passe par le pôle Nord (CIO = origine internationale conventionnelle), l'axe X passe par l'intersection entre l'équateur et le méridien de Greenwich et l'axe Y complète l'ensemble de façon à former un système d'axes cartésien direct. Le World Geodetic System 84 (WGS84) est un de ces systèmes globaux. Il est beaucoup utilisé avec les GPS et représente les valeurs en longitude, latitude, altitude.

Les systèmes de référence locaux sont des systèmes nationaux officiellement utilisés pour les mensurations d'un pays. Chaque pays possède sa propre référence géodésique et son système de projection. En Suisse, l'ellipsoïde de Bessel est utilisé. Le système de coordonnées suisse propose comme point fondamental l'ancien observatoire astronomique de Berne (coordonnées nationales 600000m/200000m).



Oracle spatial permet de faire cette transformation grâce à la méthode `sdo_transform`.

---

<sup>18</sup> [http://www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96630/sdo\\_cs\\_concepts.htm](http://www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96630/sdo_cs_concepts.htm)  
<http://topo.epfl.ch/documents/transcoco/projch/projch.php>

## Types de coordonnées

Les données cartésiennes sont des coordonnées qui mesurent la position d'un point par rapport à une origine prédéfinie le long d'axes qui sont perpendiculaires (2 ou 3 dimensions).

Les coordonnées géodésiques sont des coordonnées angulaires (longitude – latitude), liées très fortement aux coordonnées polaires sphériques.

Les coordonnées projetées sont des coordonnées cartésiennes planes qui sont le résultat d'une conversion mathématique créée à partir d'un point sur la surface de la terre.

Les coordonnées locales sont des données cartésiennes dans un système de coordonnées non géoréférencé.

## Sdo\_tolerance et arc\_tolerance<sup>19</sup>

La tolérance est utilisée pour associer un niveau de précision aux données spatiales. Elle définit la distance entre laquelle deux points peuvent être distants pour être quand même considérés comme les mêmes. La tolérance doit être un nombre positif plus grand que zéro. L'étendue des valeurs et la signification des valeurs dépendent du fait que les données spatiales soient liées à un système de coordonnées géodésique ou non.

Pour des données géodésiques, la tolérance est une valeur en mètre.

Pour des données non géodésiques, la tolérance ne peut être plus grande que 1. Elle doit être la représentation décimale de la fraction de l'unité de mesure utilisée.

Dans les deux cas, plus la tolérance est petite, plus la précision sera grande.

La tolérance est une valeur sauvegardée dans la table des métadonnées, dans la colonne diminfo.

Le mot-clé arc\_tolerance spécifie, pour chaque arc d'une géométrie, la longueur maximale de la ligne perpendiculaire entre la surface de l'arc et la ligne droite entre le point de départ et le point d'arrivée de l'arc. Cette valeur doit être plus grande ou égale à la valeur « tolérance » associée à la géométrie. L'augmentation de la valeur de arc\_tolerance fera que le polygone résultant aura moins de côtés et une plus petite surface ; si on diminue la valeur de arc\_tolerance, la polygone aura plus de côtés et une surface plus grande.

---

<sup>19</sup> [http://www.cs.uvm.edu/oracle9doc/appdev.901/a88805/sdo\\_intr.htm#884590](http://www.cs.uvm.edu/oracle9doc/appdev.901/a88805/sdo_intr.htm#884590)  
[http://dba-services.berkeley.edu/docs/oracle/manual-10gR2/appdev.102/b14255/sdo\\_objgeom.htm#i867636](http://dba-services.berkeley.edu/docs/oracle/manual-10gR2/appdev.102/b14255/sdo_objgeom.htm#i867636)



## Création de données géométriques<sup>20</sup>

Pour qu'une table soit définie comme étant une table géométrique, il faut définir plusieurs choses :

- Création de la table : la table contenant les données spatiales doit posséder au moins une colonne de type `sdo_geometry`
- Création des métadonnées : les métadonnées permettent de définir les informations relatives au tableau des dimensions ainsi que le système de références spatiales
- Création des index : Il est obligatoire de créer des index pour les colonnes géométriques pour pouvoir utiliser les fonctions proposées par Oracle Spatial
- Insertion des données spatiales

### Création de la table

```
create table route(id integer, nom varchar(25), trace mdsys.sdo_geometry) ;
```

### Création des métadonnées

```
insert into user_sdo_geom_metadata(TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
values('ROUTE', 'TRACE',
mdsys.sdo_dim_array
(mdsys.sdo_dim_element('LONG', -180.0, 180.0, 0.005),
mdsys.sdo_dim_element('LAT', -90.0, 90.0, 1)
),
8307);
```

Exemple d'insertion de metadata à 2 dimensions

Cet exemple fonctionne seulement pour une colonne « trace » 2D. Pour mettre cette colonne en 3D, il faut rajouter un élément représentant l'altitude dans `sdo_dim_array` comme présenté ci-dessous :

```
insert into user_sdo_geom_metadata(TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
values('ROUTE', 'TRACE',
mdsys.sdo_dim_array
(mdsys.sdo_dim_element('LONG', -180.0, 180.0, 0.005),
mdsys.sdo_dim_element('LAT', -90.0, 90.0, 1) ,
mdsys.sdo_dim_element('ALT', -1000.0, 1000.0, 1)
),
8307);
```

Exemple d'insertion de metadata à 3 dimensions

<sup>20</sup> OracleSpatial.pdf  
<http://lbdwww.epfl.ch/e/teaching/SlidesBDA/OracleSpatial.pdf>

## Création des index

```
create index route_geom on route(trace)
indextype is mdsys.spatial_index
parameters('layer_gtype=LINE');
```

Les fonctions spatiales pourront être utilisées seulement après avoir créer ces index sur les colonnes géométriques.

## Insertion de données

La création de données se structure ainsi :

```
create type mdsys.sdo_geometry as object(
sdo_gtype number,
sdo_srid number,
sdo_point mdsys.sdo_point_type,
sdo_elem_info mdsys.sdo_elem_info_array,
sdo_ordinates mdsys.sdo_ordinate_array) ;
```

**sdo\_gtype** : type de géométrie

format : DLTT :

- D = dimension (même que pour les métadonnées)
- L = LRS (pour notre projet 0)
- TT = type de géométrie (00 à 07)
  - o 00 = géométrie inconnue
  - o 01 = point
  - o 02 = ligne ou courbe
  - o 03 = polygone
  - o 04 = collection
  - o 05 = multipoint
  - o 06 = multiligne ou multicourbe
  - o 07 = multipolygone

**sdo\_srid** : identificateur du système de référence spatiale

- 8307 lors de l'utilisation de système de référence en longitude/latitude.

Puis selon le type de géométrie :

**sdo\_point** : point

Pour créer un point, il faut connaître ses coordonnées

- mdsys.sdo\_point\_type(x,y,z)

z peut être null si la géométrie possède 2 dimensions

Dans le cas de la création d'un point, les champs sdo\_elem\_info et sdo\_ordinates restent à null.

ou

**sdo\_elem\_info** : tableau aidant à l'interprétation des ordonnées

c'est un ensemble de triplet décrivant :

- sdo\_starting\_offset : position du premier élément
- sdo\_etype : type de l'élément (voir tableau)
- sdo\_interpretation : son interprétation (voir tableau)

ETYPE	Interpretation	Signification	Valeur et sémantique de SDO_ELEM_INFO
0		UNKNOWN_ELEMENT	
1	1	POINT	
	n	Cluster de points	
2	1	Ligne droite	
	2	Arc de cercle	
1003 ou 2003 (1:extérieur 2:intérieur)	1	Simple polygon whose vertices are connected by <u>straight line</u> segments. Specify a point for each vertex, and the last point specified must be identical to the first (to close the polygon).	
	2	Polygon made up of a connected sequence of <u>circular arcs</u> that closes on itself. The end point of the last arc is the same as the start point of the first arc.	
	3	<u>Rectangle type</u> . A bounding rectangle such that only two points, the lower-left and the upper-right, are required to describe it.	
	4	<u>Circle type</u> . Described by three points, all on the circumference of the circle.	
4	nombre d'éléments	COMPOUND LINESTRING. Eléments doivent être de type 2	
1005 ou 2005	nombre d'éléments	COMPOUND POLYGON. Eléments doivent être de type 2	

**sdo\_ordinates** : tableau de coordonnées (à faire attention aux dimensions définies dans les métadonnées)

### Exemples :

```
insert into ville values(1, 'Sion', mdsys.sdo_geometry(3001, 8307,
mdsys.sdo_point_type(46.233, 7.350, 494), null, null)) ;
```

La géométrie créée est de la forme suivante :

3001 : Point en 3 dimensions (30 = 3D, 01 = Point)

8307 : SRID → Longitude/latitude WGS84

sdo\_point\_type(x,y,z) : création d'un point de latitude x, longitude y et altitude z

Les deux champs suivants doivent rester à null lors de la création d'un point

```
insert into route values(1, 'Rue de Lausanne', mdsys.sdo_geometry(3002, 8307, null,
mdsys.sdo_elem_info_array(1,2,1), mdsys.sdo_ordinate_array(46.1,7.533,100, 46.3,
7.533, 100))) ;
```

La géométrie créée est de la forme suivante :

3002 : Ligne en 3 dimensions (30 = 3D, 02 = Ligne)

8307 : SRID → Longitude/latitude WGS84

null : ce n'est pas un point

sdo\_elem\_info\_array(1,2,1) : 1 = depuis la position 1, 2 = ligne ou courbe, 1 = ligne droite

sdo\_ordinate\_array(x1,y1,z1,x2,y2,z2,...) : coordonnées des points

Pour plus d'exemple de création de géométries voir l'annexe « Création de géométries : Exemples ».

## Index spatial basé sur des fonctions<sup>21</sup>

Il est important de noter que pour utiliser les fonctionnalités proposées par Oracle Spatial, toutes les géométries doivent posséder un index. Ainsi, pour pouvoir utiliser une géométrie retournée par une fonction définie par l'utilisateur, il faut créer un index sur cette géométrie. Cet index s'appelle un « function-based spatial index ». L'index créé est basé sur la valeur pré calculée retournée par la fonction.

Tout d'abord, il faut donner l'autorisation à l'utilisateur d'applications spatiales au privilège QUERY REWRITE.

```
grant query rewrite to user;
```

De plus, les paramètres suivants doivent être initialisés :

QUERY\_REWRITE\_ENABLED = TRUE et QUERY\_REWRITE\_INTEGRITY=TRUSTED

```
alter session set QUERY_REWRITE_ENABLED = TRUE
alter session set QUERY_REWRITE_INTEGRITY=TRUSTED
```

Ensuite, il faut suivre les points suivants :

- 1° Créer la fonction retournant une géométrie sdo\_geometry. Cette fonction doit être déclarée DETERMINISTIC. Une fonction est considérée comme déterministe si elle retourne toujours le même résultat pour les mêmes paramètres d'entrée.

```
create or replace function get_long_lat_pt(longitude in number,
                                           latitude in number)
return sdo_geometry deterministic is
begin
    return sdo_geometry(2001, 8307,
        sdo_point_type(longitude, latitude, null), null, null);
end;
```

- 2° Mettre à jour les métadonnées

```
insert into user_sdo_geom_metadata values('LONG_LAT_TABLE',
    'get_long_lat_pt(longitude,latitude)',
    sdo_dim_array(
        sdo_dim_element('Long', -180, 180, 0.005),
        sdo_dim_element('Lat', -90, 90, 0.005)), 8307);
```

- 3° Créer l'index spatial

```
create index long_lat_table_idx on
    long_lat_table(get_long_lat_pt(longitude,latitude))
indextype is mdsys.spatial_index;
```

<sup>21</sup> [http://www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo\\_exten.htm](http://www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_exten.htm)

## Utilisation de fonctions Oracle Spatial

### Buffer<sup>22</sup>

Un buffer détermine une zone arrondie autour d'une géométrie. Tous les points de cette zone se trouvent à égale distance de la géométrie d'origine.

```
select nom, mdsys.sdo_geom.sdo_buffer(a.position, b.diminfo, 2000, 'unit=m
arc_tolerance=1')
from Ville a, user_sdo_geom_metadata b
where table_name = 'VILLE' and a.id = 1;
```

Exemple de buffer autour d'un point

Avec des données géodésiques (latitude, longitude), il n'est pas possible d'utiliser des cercles pour définir une zone. Le polygone que retourne la fonction buffer est un polygone représentant au mieux un cercle grâce à des lignes droites. Il est donc logique que plus le cercle est grand, plus il y a de droite le composant.

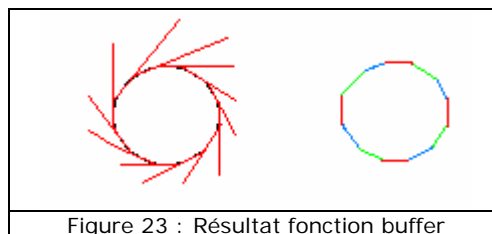


Figure 23 : Résultat fonction buffer



Figure 24 : Représentation graphique

Exemple d'un buffer de 2 km autour de Grône avec une tolérance de 1. Ici seulement 6 points sont représentés mais en réalité, le résultat du buffer retourne plus de 90 coordonnées.

<sup>22</sup> <http://forums.oracle.com/forums/thread.jspa?messageID=1093357&#1093357>  
[http://www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96630/sdo\\_cs\\_concepts.htm](http://www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96630/sdo_cs_concepts.htm)

## Filtre

Les filtres permettent de connaître les relations entre les géométries. Oracle Spatial possède deux fonctions permettant de faire des filtres : `sdo_filter` et `sdo_relate`

### *Sdo\_filter*

```
select a.id, a.nom
from chemin a
where mdsys.sdo_filter(a.trace,
    mdsys.sdo_geometry(3003, 8307, null,
    mdsys.sdo_elem_array(1, 1003, 1),
    mdsys.sdo_ordinate_array(46.09, 7.54, 0, 46.09, 7.53,
    0, 46.1, 7.52, 0, 46.29, 7.52, 0, 46.30, 7.53, 0)),
    'querytype=WINDOW') = 'TRUE';
```

Exemple de filtre : sélection de toutes les géométries (chemins) en relation avec le polygone

ID	NOM
6	Sion - Bern
5	Sierre - Bern
3	Sierre - Sierre
4	Sierre - Sion
1	Sierre - Sion
2	Sierre - Chippis

Figure 25 : Exemple de résultat

### *Sdo\_relate*

```
select a.id, a.nom
from chemin a
where mdsys.sdo_relate(a.trace,
    mdsys.sdo_geometry(3003, 8307, null,
    mdsys.sdo_elem_info_array(1, 1003, 1),
    mdsys.sdo_ordinate_array(46.09, 7.54, 0, 46.09, 7.53,
    100, 46.1, 7.52, 100, 46.29, 7.52, 100)),
    'mask=inside querytype=WINDOW') = 'TRUE';
```

Exemple de filtre : sélection des chemins se trouvant dans (inside) le polygone

ID	NOM
3	Sierre - Sierre
1	Sierre - Sion
2	Sierre - Chippis

Figure 26 : Exemple de résultat

```
select a.id, a.nom,
mdsys.sdo_geom.relate(a.trace, 'determine',
mdsys.sdo_geometry(3003, 8307, null, mdsys.sdo_elem_info_array(1, 1003, 1),
mdsys.sdo_ordinate_array(46.09, 7.54, 0, 46.09, 7.53, 100,
46.1, 7.52, 100, 46.29, 7.52, 100)), 1)
from chemin a;
```

Exemple de filtre : Affiche grâce au « determine » quel type de relation existe entre les géométries

ID	NOM	MDSYS.SDO_GEOM.RELATE()
1	Sierre - Sion	INSIDE
2	Sierre - Chippis	INSIDE
3	Sierre - Sierre	INSIDE
4	Sierre - Sion	OVERLAPBDYDISJOINT
5	Sierre - Bern	OVERLAPBDYDISJOINT
6	Sion - Bern	DISJOINT

Figure 27 : Exemple de résultat

## Coordonnées

Oracle Spatial propose la fonction « sdo\_util.getVertices » qui retourne les coordonnées d'une géométrie.

```
select g.x,g.y,g.z
from chemin c, table(mdsys.sdo_util.getVertices(c.trace)) g
where c.id = 1;
```

X	Y	Z
46,1	7,53	100
46,3	7,533	100

Figure 28 : Résultat

## Intersection

La fonction sdo\_geom.sdo\_intersection permet de retourner le ou les points d'intersection entre deux géométries.

```
select mdsys.sdo_geom.sdo_intersection(c1.trace, b.diminfo,
c2.trace, b.diminfo) as point
from chemin c1, chemin c2, user_sdo_geom_metadata b
where b.table_name = 'CHEMIN'
and b.column_name = 'TRACE'
and c1.id = 1
and c2.id = 2 ;
```

### POINT

```
SDO_GEOMETRY(3001, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 1, 1),
SDO_ORDINATE_ARRAY(46,3, 7,533, 0))
```

Figure 29 : Résultat

## Utilisation d'Oracle et de PL/SQL

---

### Procédure stockée<sup>23</sup>

Oracle ne suit pas le même schéma que SQL Server pour renvoyer des données depuis une procédure stockée. Le résultat d'une requête SELECT ne peut pas être affiché depuis la procédure stockée. Il doit être ajouté à un curseur et renvoyé. Cependant, un curseur n'est pas un type de données accepté comme paramètre de sortie d'une procédure stockée Oracle. Il faut donc créer un type de données personnalisé correspondant à une référence du curseur (ref cursor). Ref Cursor est une référence à un ResultSet se trouvant dans la mémoire du serveur. Ainsi, l'adresse du lieu où se trouvent les données est envoyée au client.

Pour utiliser ce curseur de référence, un package doit être créé. Un package est un regroupement de procédures et de fonctions. Des types personnalisés peuvent être créés pour être utilisés dans ce package. Un package se divise en deux parties distinctes : le corps qui contient l'ensemble des toutes les définitions de procédures et de fonctions et l'interface qui spécifie quelle procédures ou fonctions pourront être utilisable de l'extérieur du package.

#### *Côté Oracle*

Création d'un package avec un nouveau type : t\_cursor et une procédure : myProc

```
create or replace package myPackage as
type t_cursor is REF CURSOR ;
procedure myProc(dataOut out t_cursor) ;
end myPackage;
```

Création du corps du package. Celui-ci contient une procédure renvoyant un curseur de référence.

```
create or replace package body myPackage as
procedure myProc(dataOut out t_cursor)
is
begin
open dataOut for select * from myTable ;
end myProc ;
end myPackage ;
```

---

<sup>23</sup>

<http://www.zdnet.fr/livres-blancs/0.39035134.60246730p-39000727q.00.htm>  
[http://www.zdnet.fr/builder/programmation/windows\\_dotnet/0.39021036.213314.00.htm](http://www.zdnet.fr/builder/programmation/windows_dotnet/0.39021036.213314.00.htm)  
[http://www.infini-fr.com/Sciences/Informatique/Base\\_de\\_donnees/Relationnel/Oracle/plsql.html](http://www.infini-fr.com/Sciences/Informatique/Base_de_donnees/Relationnel/Oracle/plsql.html)



*Côté client – Visual Studio*

Ajout de la référence : `using System.Data.OracleClient;`

```
private void button1_Click(object sender, EventArgs e)
{
    //Création du DataSet
    DataSet ds = new DataSet();

    //Création et ouverture de la connexion
    OracleConnection oc = new
    OracleConnection("Server=xxx;Uid=yyy;Pwd=zzz");
    oc.Open();

    //Création de la commande et lien avec la procédure stockée Oracle
    OracleCommand ocom = new OracleCommand();
    ocom.Connection = oc;
    ocom.CommandText = "myPackage.myProc";
    ocom.CommandType = CommandType.StoredProcedure;

    //Création du paramètre de sortie et ajout à la commande
    OracleParameter pp = ocom.CreateParameter() as OracleParameter;
    pp.OracleType = OracleType.Cursor;
    pp.ParameterName = "DataOut";
    pp.Direction = ParameterDirection.Output;
    ocom.Parameters.Add(pp);

    //Appel de la procédure
    ocom.ExecuteNonQuery();

    //Création d'un DataAdapter
    OracleDataAdapter oda = new OracleDataAdapter(ocom);

    //Remplissage du DataSet avec les données
    try
    {
        oda.Fill(ds, "myTable");
    }
    catch
    {
    }

    //Affichage dans la grille
    dataGridView1.DataSource = ds.Tables[0];
    dataGridView1.Visible = true;

    //Fermeture de la connexion
    oc.Close();
}
```

Application Windows Visual Studio 2005 en C#

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        //Création du DataSet
        DataSet ds = new DataSet();

        //Création et ouverture de la connexion
        OracleConnection oc = new
        OracleConnection("Server=xx;Uid=yy;Pwd=zz");
        oc.Open();

        //Création de la commande et lien avec la procédure stockée Oracle
        OracleCommand ocom = new OracleCommand();
```

```

ocom.Connection = oc;
ocom.CommandText = "myPackage.myProc";
ocom.CommandType = CommandType.StoredProcedure;

//Création du paramètre de sortie et ajout à la commande
OracleParameter pp = ocom.CreateParameter() as OracleParameter;
pp.OracleType = OracleType.Cursor;
pp.ParameterName = "DataOut";
pp.Direction = ParameterDirection.Output;
ocom.Parameters.Add(pp);

//Appel de la procédure
ocom.ExecuteNonQuery();

//Création d'un DataAdapter
OracleDataAdapter oda = new OracleDataAdapter(ocom);

//Remplissage du DataSet avec les données
try
{
    oda.Fill(ds, "myTable");
}
catch
{
}

//Ajout du DataSet dans une Session
Session["dataset"] = ds;

//Fermeture de la connexion
oc.Close();
}

//Récupération du DataSet mis dans la Session
DataSet d = (DataSet)Session["dataset"];

//Ajout des données à la grille
GridView3.DataSource = d.Tables["myTable"];
GridView3.DataBind();
}

```

Application Web Visual Studio 2005 C#

	ID	NOM
▶	5	Sierre - Bern
	3	Sierre - Sierre
	4	Sierre - Sion
	1	Sierre - Sion
	2	Sierre - Chippis
*		

Figure 30 : Application Windows

ID	NOM
6	Sion - Bern
4	Sierre - Sion

Figure 31 : Application Web

## Processus de recherche d'itinéraires

La création dynamique d'itinéraires doit suivre un certain processus afin de filtrer les données à chaque étape. Pour optimiser l'algorithme, les premiers filtres s'occuperont de la suppression de toutes les routes non utiles à la création des itinéraires. Ainsi, les filtres demandant plus de temps d'exécution auront nettement moins de données à gérer. Ce processus se divise en 6 parties distinctes présentées ci-dessous :

### 1° Choix du lieu de départ

L'utilisateur peut choisir parmi deux types de lieu de départ : une ville (ou village) ou une commune. Il aura donc accès à toutes les routes partant de ce lieu.

### 2° Choix du rayon

Le rayon est la distance autour du lieu de départ que la personne est disposée à effectuer pour rejoindre le point de départ de la randonnée.

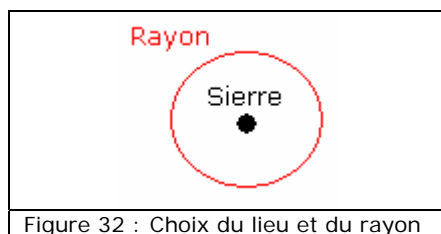


Figure 32 : Choix du lieu et du rayon

### 3° Choix du type d'activité et du type de route

La personne cherchant un itinéraire peut choisir le type d'activité qu'elle désire faire : marche à pied, vélo de route, VTT,... Chaque activité est liée à un type de route. Par exemple, un marcheur pourra aller sur n'importe quel type de route alors qu'une personne qui désire faire du vélo de route ne pourra pas emprunter des chemins caillouteux ou en forêt.

La possibilité est donnée à l'utilisateur de choisir s'il veut ou non sélectionner tous les types de routes disponibles ou seulement quelque uns.

<input type="radio"/> Vélo de route	<input type="checkbox"/> Route goudronnée
<input type="radio"/> VTT	<input checked="" type="checkbox"/> Chemin
<input checked="" type="radio"/> Marche	<input checked="" type="checkbox"/> Sentier
	<input type="checkbox"/> Cailloux

Figure 33 : Recherche d'itinéraires proposant de la marche à pied sur des chemins ou des sentiers

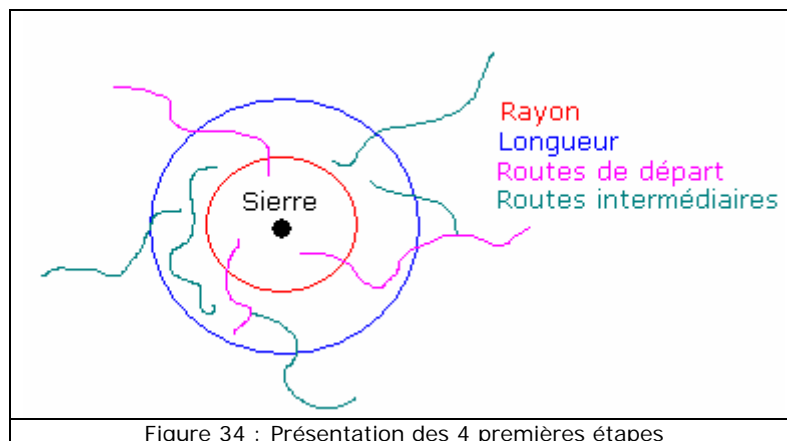
Une évolution de programme pourra être envisagée pour qu'un itinéraire nécessitant 50 mètres de routes pour passer d'un chemin à un autre soit quand même affiché.

## 4° Longueur du trajet

La longueur du trajet est importante pour supprimer une bonne partie des routes non utiles. En effet, toutes les routes débutant plus loin que la distance choisie, ne seront pas prise en considération. Cette information doit être fournie seulement si l'utilisateur n'as pas choisi de lieu d'arrivée.

### Exemple

Ces quatre premières parties, permettent de supprimer plus du 90% des routes présentes dans la base de données. L'image ci-dessous le démontre.



Prenons un exemple concret. Une personne désire faire une randonnée de 10 km à pied sur un chemin pédestre en partant de Sierre. Elle est disposée à faire 2 km pour atteindre le départ du parcours.

La zone rouge détermine toutes les routes (violet) dans un rayon de 2 km pouvant être utilisées pour le départ du trajet.

La zone bleue est une zone de 10 km autour de la zone rouge. Elle permet de supprimer toutes les routes n'ayant pas au moins une partie dans la zone. En effet, une route se trouvant à l'extérieur de la zone bleue nécessite plus de 10 km pour l'atteindre et est donc inutile pour la création de ce trajet. Il reste donc les routes ayant une partie dans la zone (routes vertes).

## 5° Type de parcours : boucle – aller simple – aller-retour

- un aller simple est un parcours dont le point de départ se trouve dans la zone rouge et se terminant dans la zone bleue
- un aller-retour est un aller simple dans les deux sens. Le point de départ est le même que le point d'arrivée
- une boucle est un parcours dont le point de départ et le point d'arrivée se trouvent dans la zone rouge, mais dont les routes de départ et d'arrivée ne sont pas similaires.
- une personne peut aussi entrer un lieu de départ (ex : Sion) et un lieu d'arrivée (ex : Sierre). Dans ce cas, le trajet sera considéré comme étant un aller simple.

Cette partie est le point central du travail de diplôme et le plus complexe à réaliser, car il faut prendre en considération les routes qui se touchent ou se croisent pour créer le parcours. Une première chose à faire est de supprimer toutes les routes vertes qui n'ont

aucun lien avec d'autres routes, puisque cela signifie qu'elles ne pourront pas être atteignables depuis une route violette.

Pour une première approche de la recherche d'itinéraire, il a été décidé que chaque route serait divisée en tronçon, c'est-à-dire, qu'une géométrie représenterait un bout de route entre 2 intersections. Plus tard, si le temps le permet, il est envisageable de revenir à une approche plus globale où des routes pourraient se croiser.

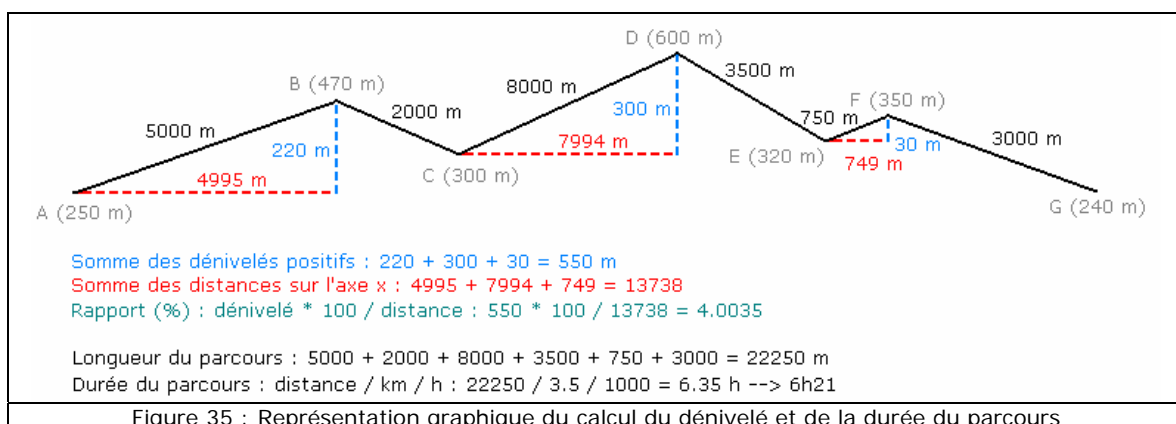
Cette partie est détaillée dans le chapitre « Graphe » à la suite de ce document.

## 6° Calcul du dénivelé, de la difficulté et de la durée du trajet

- Calcul du dénivelé : Le dénivelé se calcule en faisant la somme des dénivelés positifs du parcours. (Même chose pour les dénivelés négatifs mais lorsque les routes descendent)
- Rapport : En divisant le dénivelé positif par la longueur du trajet sur un axe x, on obtient le pourcentage d'élévation de la route.
- La difficulté est dérivée de ce dénivelé positif (table de conversion de la base de donnée)
- La durée du trajet est calculée selon la difficulté du parcours et le type d'activité effectuée, qui permettront de définir la vitesse de la personne.

Ces informations sont des moyennes mais elles permettent au randonneur de se faire une idée de ce qu'est le parcours.

### Exemple



Le rapport en pourcent permet de définir une difficulté. Partons du principe qu'une pente de 4,0035% est de difficulté moyenne. Une table de la base de données définit que pour une difficulté moyenne et une activité « marche », la vitesse correspondante est de 3.5 km/h.

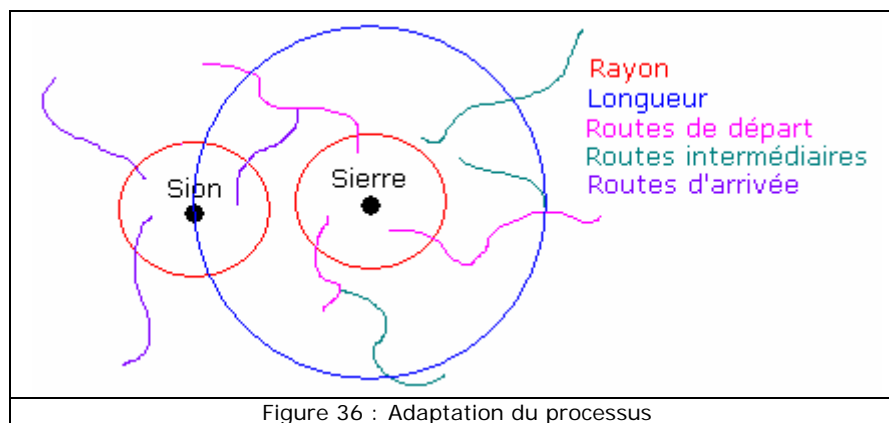
En connaissant la vitesse de la personne et la longueur du trajet, il est donc possible de calculer la durée du parcours.

## Adaptation

Le processus décrit ci-dessus est valable pour une recherche d'itinéraire dont on ne connaît pas le point d'arrivée. Cependant, le site propose également à l'utilisateur de lui fournir la liste des trajets d'un point à un autre. Dans ce cas, le processus est similaire pour les trois premières étapes mais diffère pour la quatrième partie. En effet, la distance du parcours ne peut pas être connue par la personne. Celle-ci sera donc calculée grâce à une fonction Oracle Spatial qui permet de retourner la distance entre deux géométries (ici le lieu de départ et le lieu d'arrivée). Ainsi, la zone bleue définie dans la figure 25 passera sur le lieu d'arrivée.

En plus de cela, il est nécessaire de rechercher toutes les routes partant depuis le lieu d'arrivée. Il est ensuite possible de garder les routes d'arrivée ayant une interaction avec les routes liées à la zone de départ.

Ainsi, seules les routes possédant au moins un tronçon rose et un tronçon violet seront affichées.



## Graphe

Cette partie du document explique en détail, par le biais d'un exemple, la création des différents trajets. Ce chapitre est un complément de l'étape 5 du processus de recherche d'itinéraires.

L'utilisation d'un graphe a été choisie pour représenter les différents parcours.

L'exemple se basera sur l'image présentée ci-dessous.

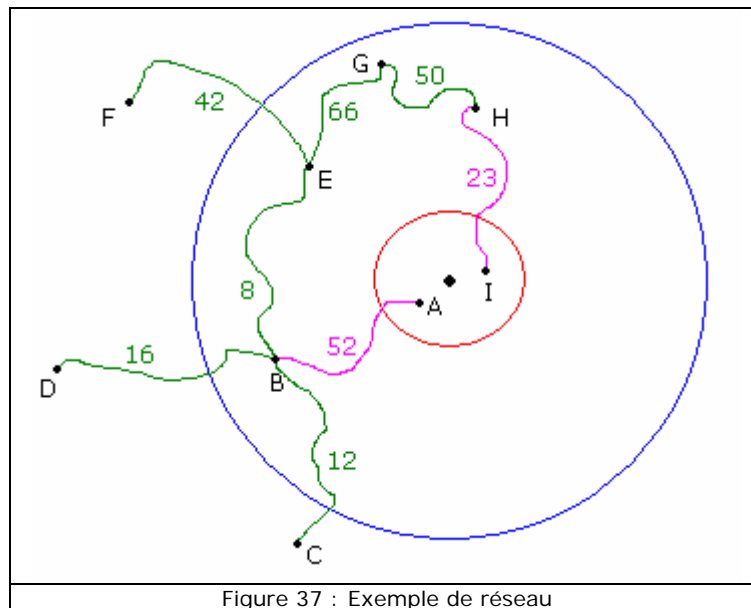


Figure 37 : Exemple de réseau

## Tables

Avant de créer le graphe représentant ce réseau, il est nécessaire de récupérer les informations de la base de données. Plusieurs tables vont donc être créées pour stocker ces données.

### Table de conversion

La table de conversion est utile pour stocker la liste de routes en leur redéfinissant un nouvel identifiant. Pour notre exemple, il ne serait pas envisageable de créer un tableau de 66 cases dont seulement 8 seraient remplies. On va donc créer un tableau de 8 cases et redéfinir le numéro de la route par son emplacement dans le tableau.

Voici un exemple de table de conversion pour notre exemple :

0	1	2	3	4	5	6	7
52	23	12	16	8	50	66	42

## Table de départ

La table de départ stocke la liste des routes de départ. On gardera les identifiants des routes selon la table de conversion.

0	1
0	1

## Table d'arrivée

Cette table stocke la liste des routes d'arrivée. Elle n'est utilisée que lors d'une recherche d'itinéraire entre deux points, c'est-à-dire lorsqu'on connaît le lieu de départ et le lieu d'arrivée d'un parcours. Pour notre exemple, cette table n'est pas nécessaire.

## Table des distances

Chaque route possède plusieurs informations importantes pour la création des itinéraires. Elles sont toutes stockées dans la table des distances. Ainsi, on gardera la distance, le dénivelé positif, le dénivelé négatif, la distance sur l'axe X (pour le dénivelé positif et le dénivelé négatif) et la durée du parcours dans les deux sens de la route. Il est important de calculer la durée dans les deux sens, car si une route monte dans un sens, elle descendra forcément si on la parcourt dans l'autre. Le temps nécessaire pour ce trajet sera donc différent.

	0	1	2	3	4	5	6	7
Distance	492	600	853	1000	976	230	300	845
Dén. pos	20	50	150	300	280	20	90	200
Dén. neg	10	30	50	250	50	20	40	250
Axe x pos	250	400	535	780	800	110	160	500
Axe x neg	242	200	218	220	176	120	140	345
Durée	0.18	0.26	0.40	0.55	0.60	0.05	0.10	0.37
Durée neg	0.15	0.13	0.16	0.14	0.15	0.09	0.07	0.31

## Tableau des points de départ

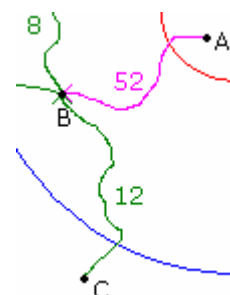
Le tableau des points de départ stocke la liste des coordonnées des points de départ de chaque route.

0	1	2	3	4	5	6	7
A	I	C	B	B	H	E	E

Ce tableau permet de savoir dans quel sens va une route, car une route démarre à son point de départ.

Ces informations sont utiles lors du calcul du dénivelé, de la durée et de la difficulté du parcours, car ces données sont différentes si on va dans un sens ou dans l'autre de la route.

Prenons l'exemple de la route 0 (n° 52) et la route 2 (n° 12). Elles se touchent mais leur sens n'est pas le même, ce qui a pour conséquence qu'elles arrivent l'une en face de l'autre. Pour calculer la durée du trajet 0 – 2, il faudra prendre la durée positive de 0 et la durée négative de 2.





## Tableau des liens

Cette table stocke la liste des liens entre chaque route. Comme pour chaque table présentée précédemment, l'identifiant utilisé est celui de conversion.

0	1	2	3	4	5	6	7	8	9	10
0	0	0	1	2	2	3	4	4	5	6
2	3	4	5	3	4	4	6	7	6	7
11	12	13	14	15	16	17	18	19	20	21
2	3	4	5	3	4	4	6	7	6	7
0	0	0	1	2	2	3	4	4	5	6

## Tableau des intersections

Le tableau des intersections permet de connaître le point qui relie deux routes. Dans notre exemple, les points sont représentés par une lettre. En réalité, ces points sont définis grâce aux valeurs x et y, respectivement la longitude et la latitude.

Ainsi,  $B = (x_B, y_B)$ .

0	1	2	3	4	5	6	7	8	9	10
B	B	B	H	B	B	B	E	E	G	E
11	12	13	14	15	16	17	18	19	20	21
B	B	B	H	B	B	B	E	E	G	E

## Graphe

### Généralité

Le graphe employé dans ce projet est un tableau de File. Cette décision a été prise car, comme le nombre d'éléments par ligne est variable, il n'est pas envisageable d'utiliser une matrice (tableau de tableau) de taille fixe.

Une File est constituée d'une série de Nœud liés entre eux. Chaque Nœud possède une Info. Pour faciliter l'accès aux Nœud, ceux-ci sont liés entre eux dans les deux sens. C'est-à-dire qu'un Nœud possède un suivant et un précédant.

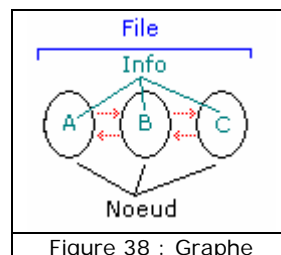


Figure 38 : Graphe

### Création du graphe

Pour pouvoir représenter les itinéraires, les Info stockées dans chaque nœud sont les suivantes :

- identifiant de la route
- identifiant de conversion
- longueur de la route
- dénivellation positive
- dénivellation négative
- durée du parcours (sens positif)
- durée du parcours (sens négatif)
- distance sur l'axe x pour le dénivelé positif
- distance sur l'axe x pour le dénivelé négatif
- coordonnée x du point de départ
- coordonnée y du point de départ
- coordonnée x du point d'intersection
- coordonnée y du point d'intersection
- route de départ ou non
- route d'arrivée ou non
- sens

La première étape de la création du graphe consiste en l'ajout d'un Nœud pour chaque route de départ. Ces routes ne possèdent pas de points d'intersection.

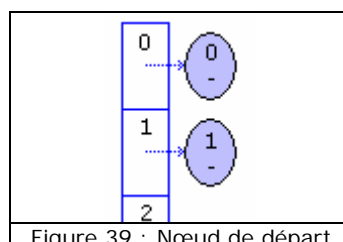


Figure 39 : Nœud de départ

Puis, on parcourt le tableau de lien et on ajoute un Nœud au graphe de la manière suivante (exemple des 2 premiers liens) :

- ajout du Nœud 2 avec point d'intersection B à la File 0
- ajout du Nœud 3 avec point d'intersection B à la File 0

0	1	0	1
0	0	B	B
2	3		
Lien		Intersection	

Si il y a un Nœud représentant une route de départ dans la liste des liens, on va mettre la valeur « départ » de ce Nœud à true

Si il y a un Nœud représentant une route d'arrivée dans la liste des liens, on va mettre la valeur « arrivée » de ce Nœud à true. (Cet attribut est renseigné seulement dans le cas d'une recherche d'itinéraires entre deux points connus)

Au final, le graphe ressemblera à cela :

0	1	2	3	4	5	6	7
0	1	0	0	0	1	4	4
-	-	B	B	B	H	E	E
492	600	492	492	492	600	976	976
52	23	52	52	52	23	8	8
2	5	3	2	2	6	5	6
B	H	B	B	B	G	G	E
853	230	1000	853	853	300	230	300
12	50	16	12	12	66	50	66
3		4	4	3		7	
B		B	B	B		E	
1000		976	976	1000		845	
16		8	8	16		42	
4				6			
B				E			
976				300			
8				66			
				7			
				E			
				845			
				42			

Ce tableau ne contient que l'identifiant, l'identifiant de conversion, le point d'intersection et la longueur du parcours. Il faut en plus ajouter le dénivelé positif, le dénivelé négatif, la durée du parcours (positive, négative), la distance sur l'axe x (positive, négative), les coordonnées du point de départ du tronçon et la valeur booléenne informant si la route est une route de départ ou une route d'arrivée.

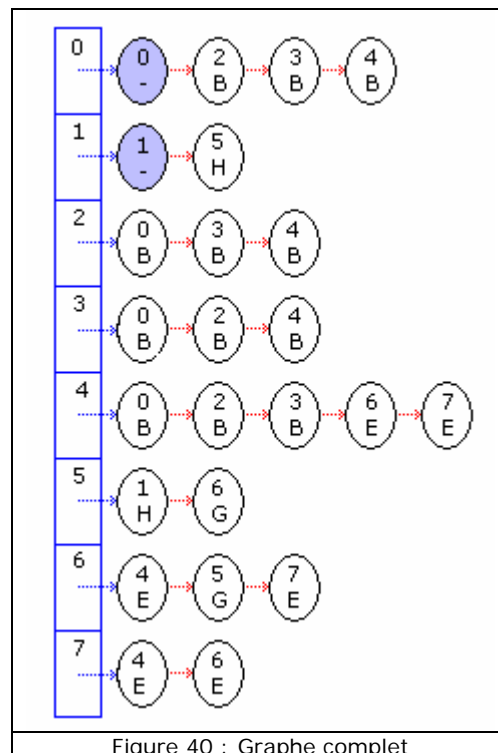


Figure 40 : Graphe complet

Représentation simplifiée du graphe avec dans chaque nœud, l'identifiant de conversion et le point d'intersection entre chaque route.

On peut voir dans le tableau ainsi que dans l'image simplifiée du graphe ci-dessus que, par exemple, la route 5 est liée à la route 1 par le point H et à la route 6 par le point G

## Recherche d'itinéraire

### Création de tous les itinéraires

A partir du moment où le graphe est créé, il faut rechercher les différents trajets possibles. Pour cela, on va parcourir le graphe ainsi :

- Une première boucle passera par chaque route de départ. (ici la 0 et la 1 → Nœud bleus). On va ajouter ce Nœud à une nouvelle Pile<sup>24</sup> qui va être elle-même ajoutée à un tableau contenant les trajets possibles.
- Ensuite, pour chaque Nœud se trouvant dans la même File que la route de départ, on va regarder si on peut l'ajouter au trajet en vérifiant que l'on ne revienne pas sur nos pas (on revient sur nos pas si le point d'intersection de 2 routes est similaire).

Exemple :

- Départ de la route 0

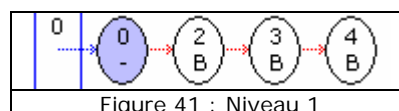


Figure 41 : Niveau 1

<sup>24</sup> Une pile est semblable à une File mais les Nœud sont empilés les uns sur les autres. Lorsque l'on dépile une Pile, c'est le Nœud ajouté en dernier qui est supprimé.

- On prend le premier Nœud de la File 0 (route 0) et on le met dans une Pile. Cette Pile est ajoutée à la liste des chemins possible.

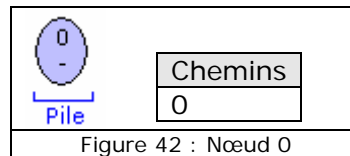


Figure 42 : Nœud 0

- On prend le 2<sup>e</sup> Nœud de cette File et on regarde si le point d'intersection est similaire à celui du Nœud précédent.

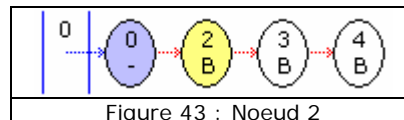


Figure 43 : Nœud 2

Ici on voit que le 2<sup>e</sup> Nœud a B comme point d'intersection et que le 1<sup>e</sup> Nœud n'en a pas. Donc on peut avancer. Un ajout du Nœud est fait dans la Pile puis la Pile est ajoutée à la liste.

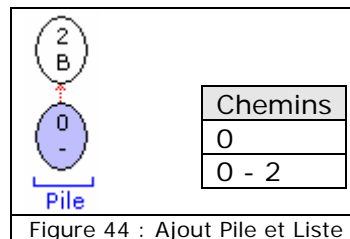


Figure 44 : Ajout Pile et Liste

- ensuite, par récursivité (monte d'un niveau), on va regarder si il existe un chemin après la route 2. Pour cela, on va passer dans la File 2 et regarder si le point d'intersection entre les éléments de cette File et le point d'intersection de la route 2 (B) sont similaires.

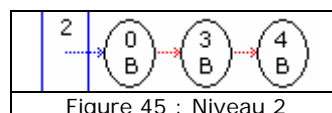


Figure 45 : Niveau 2

Dans l'exemple suivant, on remarque que la route 2 est liée à la 0, à la 3 et à la 4 mais à chaque fois par le point B, ce qui signifie qu'on revient sur nos pas. On ne va donc pas prendre en considération ces routes et redescendre d'un niveau.

- Redescendre d'un niveau consiste à revenir sur la route initiale et à prendre le Nœud suivant. Il faut aussi supprimer le dernier Nœud de la Pile. Le même processus est repris sur tous les Nœud de la File jusqu'à ce qu'on arrive à la fin.

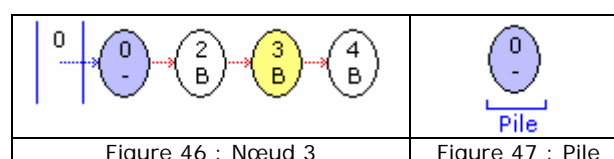


Figure 46 : Nœud 3

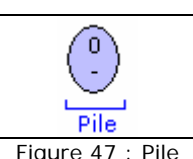


Figure 47 : Pile

Au final, le graphe créé nous renverra les chemins suivants :

Chemin	
0	1
0 – 2	1 – 5
0 – 3	1 – 5 – 6
0 – 4	1 – 5 – 6 – 4
0 – 4 – 6	1 – 5 – 6 – 4 – 0
0 – 4 – 6 – 5	1 – 5 – 6 – 4 – 2
0 – 4 – 6 – 5 – 1	1 – 5 – 6 – 4 – 3
0 – 4 – 7	1 – 5 – 6 – 7

### Choix des chemins

Selon le type de parcours choisi par l'utilisateur (boucle, aller simple ou aller-retour), les chemins retenus ne seront pas les mêmes.

Pour les boucles, on ne prendra que les chemins dont le tronçon d'arrivée se trouve dans la zone de départ (zone rouge).

Dans notre exemple, les chemins suivants seront conservés :

Chemin – Boucle	
0 – 4 – 6 – 5 – 1	1 – 5 – 6 – 4 – 0

Tous les autres chemins seront présents pour des allers simples ou des allers-retours :

Chemin – Aller simple et aller-retour	
0	1
0 – 2	1 – 5
0 – 3	1 – 5 – 6
0 – 4	1 – 5 – 6 – 4
0 – 4 – 6	1 – 5 – 6 – 4 – 2
0 – 4 – 6 – 5	1 – 5 – 6 – 4 – 3
0 – 4 – 7	1 – 5 – 6 – 7

Remarque :

Lors de la création du tableau des chemins aller-retour, les informations subissent des traitements supplémentaires entre autres sur la distance, le dénivelé et la durée. En effet, la distance de chaque tronçon sera doublée, le dénivelé positif devient le dénivelé positif plus le dénivelé négatif et la durée devient la durée positive plus la durée négative.

Pour la recherche d'itinéraire entre deux points précis, on fait une recherche de chemin aller simple et on garde seulement les trajets dont le dernier tronçon est un chemin d'arrivée.

*Traitement des itinéraires*

Il faut ensuite transformer ces tableaux de File et les stocker dans une DataTable. Pour chaque chemin, il faut calculer sa distance, son dénivelé positif, son dénivelé négatif, sa durée et sa difficulté en faisant une boucle sur tous les Nœud de la File. La difficulté est trouvée en faisant un rapport entre le dénivelé et la distance sur l'axe x.

Exemple de Table représentant les chemins « Boucle » :

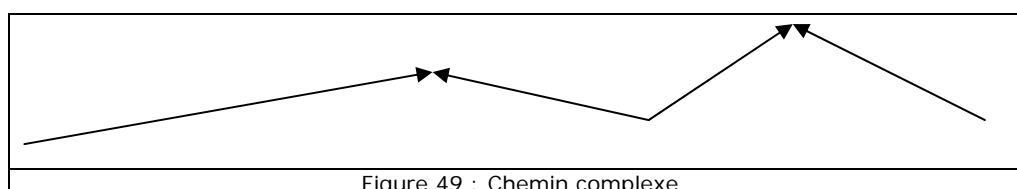
Nom	Distance	Dénivelé positif	Dénivelé négatif	Durée	Difficulté
<u>0,4,6,5,1</u>	2598	460	150	1h00	Facile
<u>1,5,6,4,0</u>	2598	150	460	1h00	Facile

Il faudra pour chaque tronçon de route prendre en considération son sens selon le sens du parcours.

Par exemple, dans le chemin suivant, le calcul du dénivelé positif se fait en additionnant ceux de tous les tronçons :



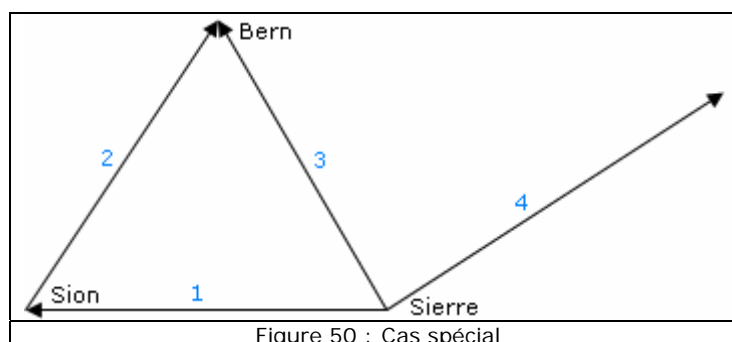
Par contre, dans le chemin ci-dessous, il n'est pas possible de faire la somme des dénivelés positifs. Il faudra ici, faire la somme des dénivelés positifs du tronçon 1 et 3 et ajouter la somme des dénivelés négatifs des tronçons 2 et 4.



Le même problème apparaît pour la durée et pour la difficulté du parcours.

Remarque :

Lors de la sélection des trajets « Départ-Arrivée », il faut faire attention à un cas de figure spécial.



Ici, on veut aller de Sierre à Bern. Il y a donc 2 parcours possibles : 3 ou 1-2. Pour trouver ces parcours, on regarde quels sont parmi tous les parcours possibles lesquels ont une route d'arrivée comme dernier Nœud. Seulement, comme route d'arrivée, on a les routes 2 et 3. Si on ne fait pas de tests supplémentaires, on aura le trajet 3-2 qui sera aussi pris en compte car 2 est un chemin d'arrivée. Il faut donc supprimer les trajets dont 2 tronçons d'arrivée se suivent.

### *Elimination des trajets trop longs*

Finalement, les trajets étant plus long que la distance choisie par l'utilisateur ne seront pas affichés.



## Interface « Exportation »

La recherche d'itinéraires a été la plus grande partie de ce travail. Cependant, le seul affichage des résultats n'est pas très attractif pour un utilisateur du site web. C'est pourquoi, le site propose également une représentation graphique de l'itinéraire ainsi que la possibilité de télécharger un fichier contenant les coordonnées des trajets, intégrable à un GPS.

Ces fonctionnalités nécessitent la création de fichiers sur le serveur. La représentation graphique se fait grâce à l'API GoogleMaps et crée un fichier htm. Les coordonnées GPS sont exportées vers un fichier au format GPX.

Une interface d'exportation a donc été mise en place afin de donner une structure équivalente à chaque classe d'objet implémentant cette interface. Ainsi, il sera possible par la suite d'ajouter facilement de nouveaux formats d'exportation, puisque la signature des méthodes principales sera déjà définie.

```
interface Exportation
{
    /**
     * Création du fichier
     */
    void creerFichier(DataTable dtRoute, DataTable dtInteret);

    /**
     * Header
     */
    string header(DataTable dtRoute, DataTable dtInteret);

    /**
     * Body
     */
    string body();
}
```

Les deux prochains chapitres expliquent le processus à suivre pour créer des cartes GoogleMaps ainsi que des fichiers contenant des données GPS.

## API GoogleMaps <sup>25</sup>

L'API de Google Maps permet d'afficher des cartes dans des pages web grâce à du JavaScript. Il est possible de rajouter des informations sur ces cartes, telles que des polygones, des marqueurs et des informations sous forme d'info-bulle.

Cet API est gratuit et il suffit de s'inscrire pour avoir une clé à insérer dans les pages web. Il faut cependant faire attention, car cette clé n'est valable que pour un répertoire du site. Ainsi, si on crée une clé pour le site [www.mysite.com/directory/](http://www.mysite.com/directory/), elle sera valide pour des url comme :

[www.mysite.com/directory/mypage.html](http://www.mysite.com/directory/mypage.html)  
[www.mysite.com/directory/page?arg=foo](http://www.mysite.com/directory/page?arg=foo)

par contre, les pages suivantes ne pourront pas afficher de cartes :

[zzz.mysite.com/directory/](http://zzz.mysite.com/directory/)  
[www.mysite.com/directory/subdir/page.html](http://www.mysite.com/directory/subdir/page.html)

### Fichier de base

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
  <head>
    <meta http-equiv="Content-Language" content="fr-ch">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title>Google Maps JavaScript API Example</title>

    <!-- Clé pour l'utilisation de la carte-->
    <script src="http://maps.google.com/maps?file=api&v=2&key=xyz"
      type="text/javascript"></script>

  </head>

  <body onload="load()" onunload="GUnload()">
    <!-- Affichage de la carte-->
    <div id="map" style="width: 500px; height: 300px"></div>
  </body>
</html>
```

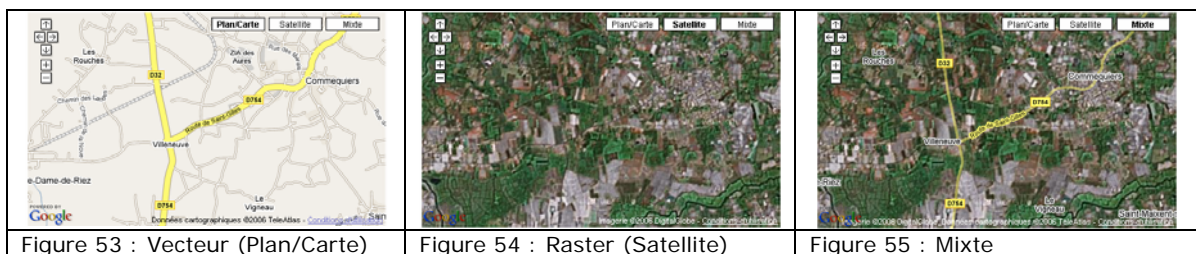
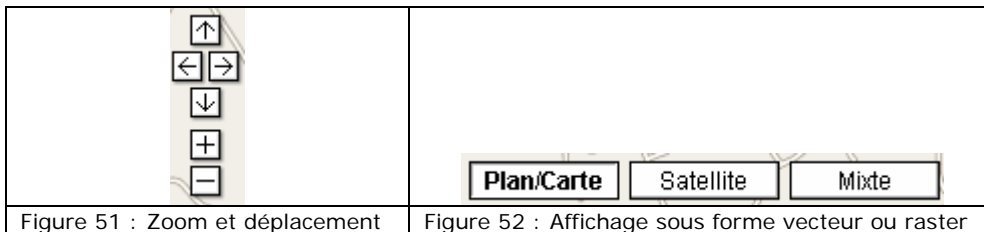
Base d'un fichier HTML pour l'affichage des cartes Google Maps

<sup>25</sup> Google Maps API  
<http://www.google.com/apis/maps/>

## Affichage d'une carte

L'affichage d'une carte ne nécessite que la création d'une instance Map et le point central de la carte à afficher sous forme longitude – latitude.

On peut y ajouter les outils de zoom, de déplacement et d'affichage.



```

<!--Affichage d'une carte centrée sur un point -->
<script type="text/javascript">
//
function load()
{
    //Test pour savoir si le navigateur web est compatible
    if (GBrowserIsCompatible())
    {
        //création d'une carte
        var map = new GMap2(document.getElementById("map"));

        //ajout des contrôles (déplacement, zoom)
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());

        //centrer la carte sur le point (46.71, -1.80) avec un zoom à 13
        map.setCenter(new GLatLng(46.71, -1.80), 13);
    }
}
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="317 731 677 746" data-label="Text">
<p>A ajouter dans le &lt;head&gt; après le script pour la clé</p>
</div>
<div data-bbox="113 930 270 957" data-label="Page-Footer">
<p>Soloz Nathalie<br/>HEVs, décembre 2006</p>
</div>
<div data-bbox="858 930 888 945" data-label="Page-Footer">
<p>69</p>
</div>
```

## Poly Ligne

Pour afficher un itinéraire en surligné, il suffit de créer un vecteur de points et d'y insérer les points sous forme latitude – longitude, puis des les afficher en polyline.

```
//création d'un vecteur de points
var points = [];

//ajout de points dans le vecteur
points.push(new GLatLng(46.699321907, -1.763768983)) ;
points.push(new GLatLng(46.699754705, -1.764260141)) ;
points.push(new GLatLng(46.700655828, -1.765507074)) ;

//affichage de la ligne formée par les points
map.addOverlay(new GPolyline(points));
```

Affichage du tracé en surligné



Figure 56 : Carte avec tracé surligné

## Marqueurs (balises)

Les balises permettent de faire ressortir un endroit particulier. En cliquant sur ces marqueurs, il est possible de faire afficher un message dans une info-bulle.

```
//ajout d'un marqueur sur le point (46.71, -1.80)
var marker = new GMarker(new GLatLng(46.71, -1.80)) ; //création d'un marker

//ajout d'un listener au marqueur pour qu'un message s'affiche quand on clique
GEvent.addListener(marker, "click", function(){
//Texte à afficher
marker.openInfoWindowHtml("<font size=1 face=Verdana>Départ</font>"));

//affichage du marqueur
map.addOverlay(marker);
```

Ajout d'un marqueur avec info-bulle



Figure 57 : Marqueur et Info-bulle

## Marqueurs personnalisés

L'API Google Maps permet de personnaliser les marqueurs. Il est possible de modifier l'image et son ombre.

```
//pour mettre mes propres markers
var baseIcon = new GIcon();
baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png"; //image ombre
baseIcon.iconSize = new GSize(20, 34); //taille de l'image
baseIcon.shadowSize = new GSize(37, 34); //taille de l'ombre
baseIcon.iconAnchor = new GPoint(9, 34);
baseIcon.infoWindowAnchor = new GPoint(9, 2);
baseIcon.infoShadowAnchor = new GPoint(18, 25);

//création d'un point
var point = new GLatLng(46.699321907,-1.763768983);

//création d'une icône de base
var icon = new GIcon(baseIcon);

//ajout de l'image à l'icône
icon.image = "http://www.google.com/mapfiles/markerD.png";

//création d'un marqueur contenant un point et un icône personnalisée
var marker = new GMarker(point,icon) ;

//Listener
GEvent.addListener(marker, "click", function(){
marker.openInfoWindowHtml("<font size=1 face=Verdana>Départ</font>"));

//affichage du marqueur
map.addOverlay(marker);
```

Ajout de marqueurs personnalisés



Figure 58 : Carte avec balise personnalisée

Cet exemple montre juste le même marqueur proposé par Google Maps mais avec un « D » inscrit dessus.

Pour plus de détails, le site de Google Maps API propose de nombreux exemples.

De plus, un exemple complet (GoogleMaps exemple complet.htm) contenant deux trajets et un marqueur personnalisé se trouve sur le DVD dans le répertoire « Documents/Exemples/ ».

## GPS

### Fonctionnement d'un GPS

GPS<sup>26</sup> signifie Global Positioning System (ou en français, système de positionnement mondial). Il s'agit du nom du principal système de positionnement par satellite mondial actuel. Il est en ce moment le seul à être entièrement opérationnel.

Le GPS a été mis en place vers la fin des années 70 par le Département de la Défense des Etats-Unis pour permettre à tous les éléments de l'armée américaine (avions, navires, troupe,...) de se positionner précisément n'importe où sur la terre. Le GPS utilise le système géodésique WGS84, auquel se réfèrent les coordonnées calculées grâce au système.

Le système GPS est composé de 24 satellites artificiels orbitant à plus de 20000 km de la terre. A tout instant, au moins 3 d'entre eux sont « visibles » de n'importe quel point de la terre. Ces satellites émettent constamment un signal codé contenant des informations dont notamment sa position et l'heure de l'émission du signal.



Un récepteur GPS peut capter ces signaux et, en effectuant des mesures, connaître sa distance par rapport aux satellites. Puis, le GPS peut situer en trois dimensions n'importe quel point (avec une précision de 15 à 100 mètres). Pour pouvoir récolter ses informations, il faut 4 satellites. Un fournit la latitude, un autre la longitude, le 3<sup>e</sup> l'altitude et le dernier le décalage temporel entre l'horloge récepteur et les satellites.

### Fonctions embarquées sur les récepteurs GPS <sup>27</sup>

**Position :** Chaque GPS positionne une personne ou un objet au minimum par rapport à sa latitude et sa longitude, mais d'autres systèmes de coordonnées peuvent être aussi trouvés (UTM, Lambert, Grille,...) selon les modèles. La précision des résultats dépendra du nombre de satellite que le GPS capte.

**Altitude :** Les GPS donne également l'altitude mais avec une précision trois fois inférieure à celle de la latitude et de la longitude. Certains récepteurs sont tout de même équipés d'un altimètre barométrique pour affiner l'affichage de l'altitude (5 m).

**Boussole :** Tous les GPS ont une fonction boussole mais qui ne fonctionne généralement qu'en « mouvement ». Il faut choisir des modèles plus perfectionnés disposant d'un compas magnétique pour pouvoir utiliser une boussole à l'arrêt.

**Liaison PC :** Les GPS actuels possèdent tous une liaison PC permettant d'importer et d'exporter des points. La norme sur les appareils de navigation établie par la NMEA préconise une liaison par port série, mais il existe des adaptateurs pour un branchement sur le port USB.

**Gestion des waypoints, traces et routes :** La différence entre les modèles réside dans le nombre de points et de traces stockables dans la mémoire.

**Cartographie :** Certains GPS intègrent un système de cartographie embarqué comme GpsTopo. Seul quelques éléments caractéristiques sont représentés.

<sup>26</sup> [http://www.ac-orleans-tours.fr/svt/SIG/dossiers/sig\\_local/HTML/page3b.htm](http://www.ac-orleans-tours.fr/svt/SIG/dossiers/sig_local/HTML/page3b.htm) et Wikipedia

<sup>27</sup> [http://www.skitour.fr/articles/read\\_78.html](http://www.skitour.fr/articles/read_78.html)

## Format

Les données GPS sont actuellement disponibles dans une multitude de formats, puisque pratiquement chaque programme et outil possède son propre format. Malgré cela, le format GPX XML (.gpx) est en passe de devenir un standard. La plus part des GPS et logiciels de traçage propose maintenant d'exporter ou d'importer des fichiers GPX. Cependant, il existe un exécutable gratuit GpsBabel qui permet une conversion entre les différents types de fichiers. Une version « en ligne » de ce logiciel existe sur le site GpsVisualizer<sup>28</sup>.

Ce document va quand même parler des principaux formats GPS.

### GPX <sup>29</sup>

GPX ou GPS eXchange Format est un format de données XML allégé mis en place pour transférer des données GPS entre applications et services web. Il peut être utilisé pour décrire des waypoint (point avec un nom et éventuellement un commentaire), des routes (ensemble de waypoints ordonnés) ou des traces (itinéraire sans waypoint).

GPX est depuis sa première sortie en 2002, le standard XML de-facto pour l'échange de données GPS.

GPX est actuellement supporté par beaucoup de logiciels gérant des données GPS ou des SIG.

Exemple de fichier GPX :

```
<?xml version="1.0" encoding="UTF-8" ?>
<gpx version="1.0" creator="CartoExploreur 3 3.08 - http://www.bayo.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.topografix.com/GPX/1/0"
  xsi:schemaLocation="http://www.topografix.com/GPX/1/0
http://www.topografix.com/GPX/1/0/gpx.xsd">
  <time>2006-04-21T15:10:22Z</time>
  <trk>
    <name>09 Coex St Gilles</name>
    <trkseg>
      <trkpt lat="46.697702010" lon="-1.762467570">
        <ele>47.000000</ele>
      </trkpt>
      <trkpt lat="46.697926621" lon="-1.762484614">
        <ele>47.000000</ele>
      </trkpt>
      <trkpt lat="46.698525039" lon="-1.763381142">
        <ele>46.000000</ele>
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```

Cet exemple décrit une trace, nommée « Coex St Gilles ». Cette trace possède un segment contenant 3 points. Ces points sont constitués d'une longitude, d'une latitude et d'une élévation (altitude). (Dans le DVD, dossier « Documents/Sources externes/GPX/ se trouve le fichier GPX.xsd qui correspond au schéma XML d'un document .gpx)

<sup>28</sup> [http://www.gpsvisualizer.com/gpsbabel/gpsbabel\\_convert?all\\_types=0&intype=gpx&outtype=mx&type=w&zip\\_output=0](http://www.gpsvisualizer.com/gpsbabel/gpsbabel_convert?all_types=0&intype=gpx&outtype=mx&type=w&zip_output=0)

<sup>29</sup> <http://www.topografix.com>

PCX5<sup>30</sup>

Ce format a été créé par Garmin, leader dans la création d'instruments GPS. Les fichiers PCX5 sont des fichiers textes, pouvant donc facilement être modifiés. Ils ont comme extension .trk (traces), .wpt (waypoint) et .rte (route).

Actuellement, ce format n'est plus supporté par Garmin mais il est toujours présent sur le marché.

Exemple de fichier .rte :

H	SOFTWARE NAME & VERSION								
I	PCX5 2.09								
H	R	DATUM	IDX	DA	DF	DX	DY	DZ	
M	G	WGS 84	121		+0.000000e+00	+0.000000e+00	+0.000000e+00	+0.000000e+00	
H	COORDINATE SYSTEM								
U	LAT LON DM								
R	00								
H	IDNT	LATITUDE	LONGITUDE	DATE	TIME	ALT	DESCRIPTION	PROXIMITY	SYMBOL ;waypts
W	HOME	N3519.37760	W08540.45030	24-MAR-94	02:00:00	-9999	MY HOUSE	1.00000e+03	10
W	FRED	N3517.78160	W08605.05230	24-MAR-94	02:00:00	-9999	FREDS HOUSE	1.00000e+04	10
W	MARINA	N3510.68190	W08536.03330	24-MAR-94	02:00:00	-9999	GOOD FOOD HERE	2.00000e+02	0
W	FCOVE	N3459.38160	W08527.88650	24-MAR-94	02:00:00	-9999	FISHING COVE	2.50000e+03	7
R	01								
H	IDNT	LATITUDE	LONGITUDE	DATE	TIME	ALT	DESCRIPTION	PROXIMITY	SYMBOL ;waypts
W	FCOVE	N3459.38160	W08527.88650	24-MAR-94	02:00:00	-9999	FISHING COVE	2.50000e+03	7
W	MARINA	N3510.68190	W08536.03330	24-MAR-94	02:00:00	-9999	GOOD FOOD HERE	2.00000e+02	0
W	FRED	N3517.78160	W08605.05230	24-MAR-94	02:00:00	-9999	FREDS HOUSE	1.00000e+04	10
W	HOME	N3519.37760	W08540.45030	24-MAR-94	02:00:00	-9999	MY HOUSE	1.00000e+03	10
R	02								
H	IDNT	LATITUDE	LONGITUDE	DATE	TIME	ALT	DESCRIPTION	PROXIMITY	SYMBOL ;waypts
W	HOME	N3519.37760	W08540.45030	24-MAR-94	02:00:00	-9999	MY HOUSE	1.00000e+03	10
W	MARINA	N3510.68190	W08536.03330	24-MAR-94	02:00:00	-9999	GOOD FOOD HERE	2.00000e+02	0
W	FRED	N3517.78160	W08605.05230	24-MAR-94	02:00:00	-9999	FREDS HOUSE	1.00000e+04	10
W	FCOVE	N3459.38160	W08527.88650	24-MAR-94	02:00:00	-9999	FISHING COVE	2.50000e+03	7

Figure 59 : Exemple fichier PCX5

Ce fichier stocke 3 routes (rouge) composées chacune de 4 points (bleu). Chaque point possède un identifiant, une latitude, une longitude, une date ainsi que l'heure de saisie, une altitude, une description, une proximité ainsi qu'un symbole. Le symbole est en fait utile seulement dans un GPS qui pourra afficher une icône sur la carte, à l'emplacement du point.

(37) 5  Boat Ramp	(63) 8  Exit	(57) 13  Hunting Area	(19) 7  Shipwreck
(38) 11  Campground	(7) 4  Fishing Area	(0) 6  Marina	(14) 9  Skull and Crossbones
(56) 3  Car	(64) 10  Flag	(43) 14  Medical Facility	(67) 15  TracBack Point
(65) 12  Circle with X	(8) 2  Gas Station	(10) 1  Residence	(18) 0  Waypoint

Figure 60 : Liste non exhaustive des symboles disponibles dans les GPS

<sup>30</sup> <http://home.online.no/~sigurdhu/MapSource-text.htm>



## NMEA 0183<sup>31</sup>

C'est une spécification combinant des instruments électriques et des données pour permettre la communication entre GPS. Elle a été mise en place par la US Based National Marine Electronics Association.

Ce standard utilise un protocole de communication simple, basé sur ASCII, définissant la façon dont sont transmises les données d'un émetteur à un receveur. Il définit aussi le contenu de chaque « phrase » :

- commence par \$
- les 5 caractères suivants identifient le type de message
- données
- checksum de 2 digits précédé d'un astérisque (\*)

Exemple :

```
$GPAAM,A,A,0.10,N,WPTNME*43
```

```
GPAAM : Alarme d'arrivée  
A : Entrée dans le cercle  
d'arrivée  
A : Perpendiculaire  
0.10 : Radius  
N : Miles nautiques  
WPTNME : Nom du waypoint  
*43 : checksum
```

## TXT

Il est aussi possible de représenter des données GPS sous forme de fichier texte avec séparateur (tabulation). Ce format est facilement intégrable à des tableurs ou des bases de données (MS Excel, MS access, Mysql...), cependant il pose plus de problème pour l'intégration à des GPS ou à des logiciels SIG.

<sup>31</sup> <http://www.kh-gps.de/nmea-faq.htm>

## Intégration des données

Pour illustrer au mieux ce travail, il aurait été intéressant d'avoir des données réelles et précises sur les divers chemins, routes, villes et communes du Valais. Ainsi, plusieurs sources de données ont été recherchées auprès de différents services de l'Etat du Valais.

Un premier contact a été pris avec le service de routes et cours d'eau, puis un deuxième avec le service des mensurations cadastrales.

### Strada-DB<sup>32</sup>

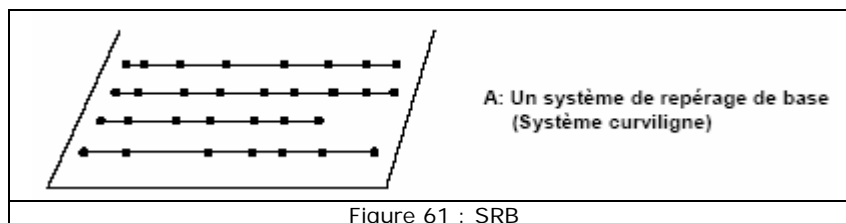
#### Présentation

Le Service des Routes et Cours d'eau de l'Etat du Valais possède une base de données appelée Strada-DB. Cette base est établie selon la norme VSS sur le repérage des données routières éditée par l'Union des professionnels suisses de la route.

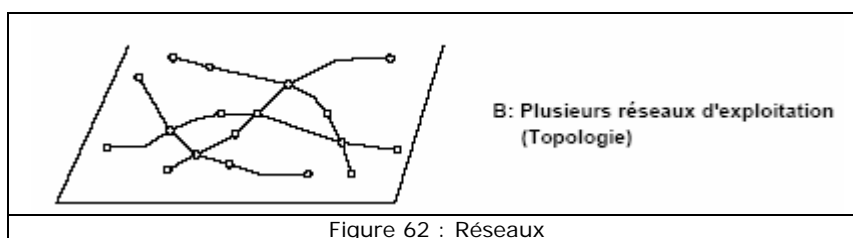
Il existe en Suisse plusieurs bases de données cantonales Strada-DB étant compatibles avec une base centralisée au niveau fédéral. Elle est le noyau commun entre les bases de données routières des cantons, de l'Office Fédéral des routes ainsi que d'autres services cantonaux et fédéraux.

#### Structure des données

Le positionnement spatial des routes est exprimé dans un système curviligne appelé Système de Repérage de Base (SRB). Celui-ci permet de repérer les divers objets (ou lieux) présents sur les routes indépendamment de la topologie.



Pour définir la topologie des routes, des réseaux d'exploitation sont rattachés au SRB. Ainsi, il est possible de créer des couches selon les processus métiers nécessaires à l'utilisateur, comme la description d'itinéraire ou la représentation de règlements (charge du trafic, réseau de déneigement,...).



<sup>32</sup>Les informations relatives à Strada-DB ont été récoltées lors de la séance avec M. Yves Molk du service des routes et cours d'eau de l'Etat du Valais et M. Pierre Gonin d'INSER ainsi que sur le document « Normalisation.pdf » du projet syrrou sur <http://topo.epfl.ch/siroute/syrrou/documents.html>

Finalement, pour lier ces deux niveaux, ensemble et au système cartésien de coordonnées nationales, un troisième niveau est ajouté, les géométries. Ce niveau fournit les éléments géométriques permettant d'établir des représentations cartographiques de la route et de ses objets associés.

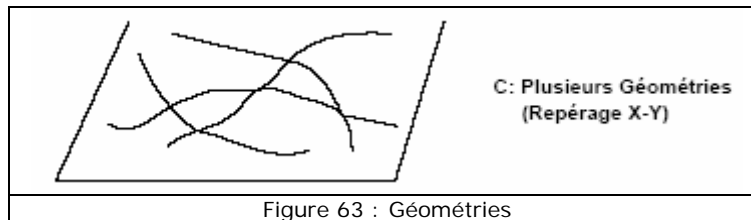


Figure 63 : Géométries

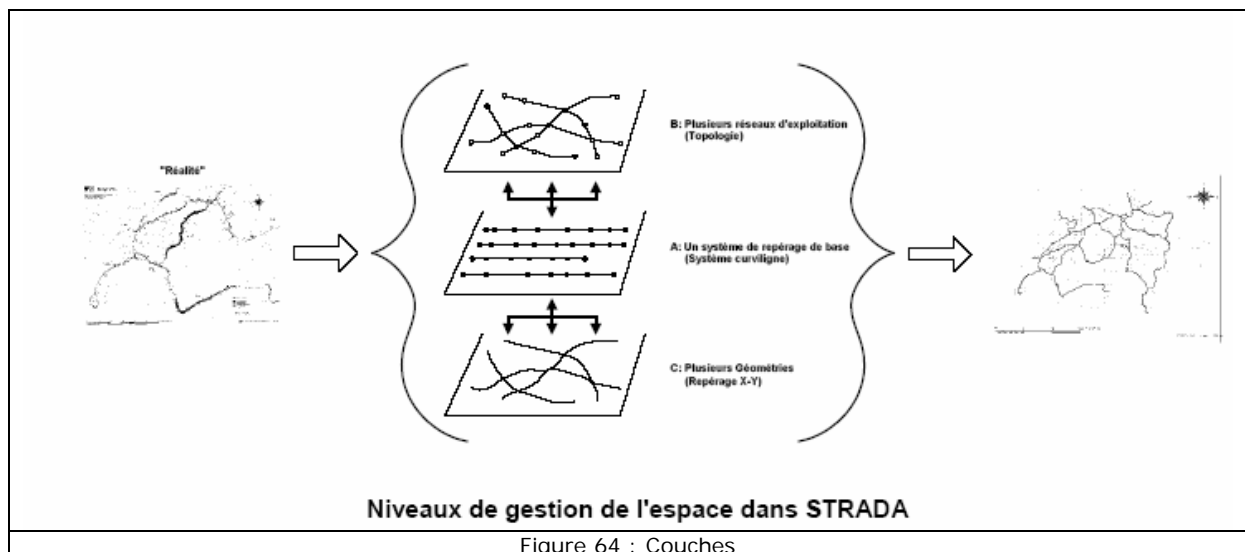


Figure 64 : Couches

Au départ, Strada-DB n'a pas été développé pour représenter les routes mais pour pouvoir analyser le positionnement de « lieux » (accident, panneau,...) par rapport à des points de repère situés sur la route, tous les kilomètres. La partie géométrie a été conçue pour répondre au besoin de l'Etat du Valais, sans utiliser de base de données spatiales.

Le point suivant explique comment sont gérées les géométries dans la base Strada-DB et pourquoi celles-ci ne correspondent pas vraiment au besoin pour ce projet.

### Description des géométries

Dans Strada-DB, une route est découpée et est représentée par une série de lignes droites. Pour chaque segment de la route, on connaît son point de départ, sa longueur (L) et son azimuth (angle par rapport à l'axe Nord), ce qui permet de le représenter graphiquement grâce à des calculs. Il n'y a donc aucune utilisation de données géographiques en tant que tel mais plutôt des données standard gérées comme des données spatiales.

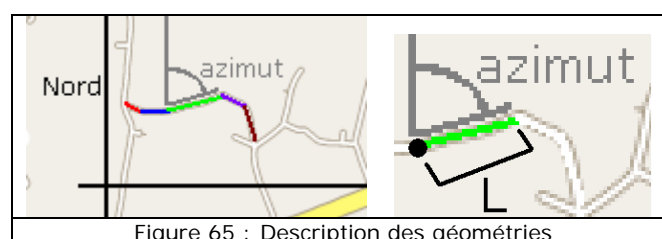


Figure 65 : Description des géométries

Pour passer de cette représentation des géométries à une représentation OpenGIS des données spatiales, beaucoup trop de transformations sont nécessaires. De plus, l'altitude, information importante pour ce projet, n'est pas gérée par Strada-DB. D'autres données ont donc été recherchées.

## Données des mensurations cadastrales<sup>33</sup>

### Présentation

Les mensurations cadastrales regroupent toutes les données géographiques de l'Etat du Valais. Il a donc été possible d'avoir accès à une série d'informations, telles que les chemins pédestres, les axes routiers, les communes et les pistes cyclables. Toutes ces données vectorielles sont sous format ShapeFile.

### Structure des données

L'utilisation d'un utilitaire gratuit, Shape Viewer 1.2, permet de visualiser les données stockées dans le ShapeFile.

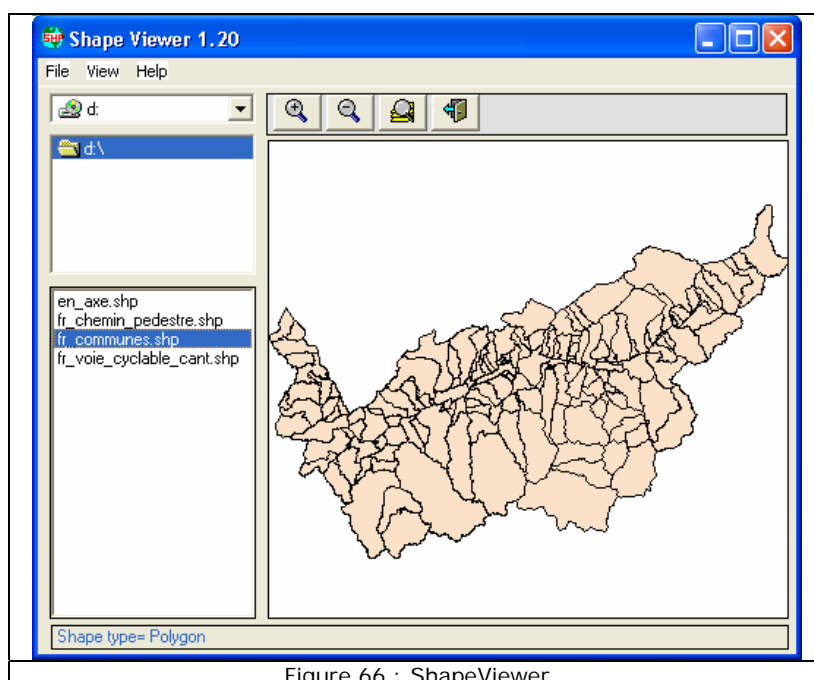


Figure 66 : ShapeViewer

Cet utilitaire permet également d'exporter les données sous Excel.

Polygon Id	Part Id	Point Id	X	Y
1	1	1	569154.384	96903.9
1	1	2	569180	96913
1	1	3	569198	96925
1	1	4	569225	96983
1	1	5	569243	96998
1	1	6	569293	97018
1	1	7	569313	97038

Le fichier ainsi créé fait plus de 65000 lignes.

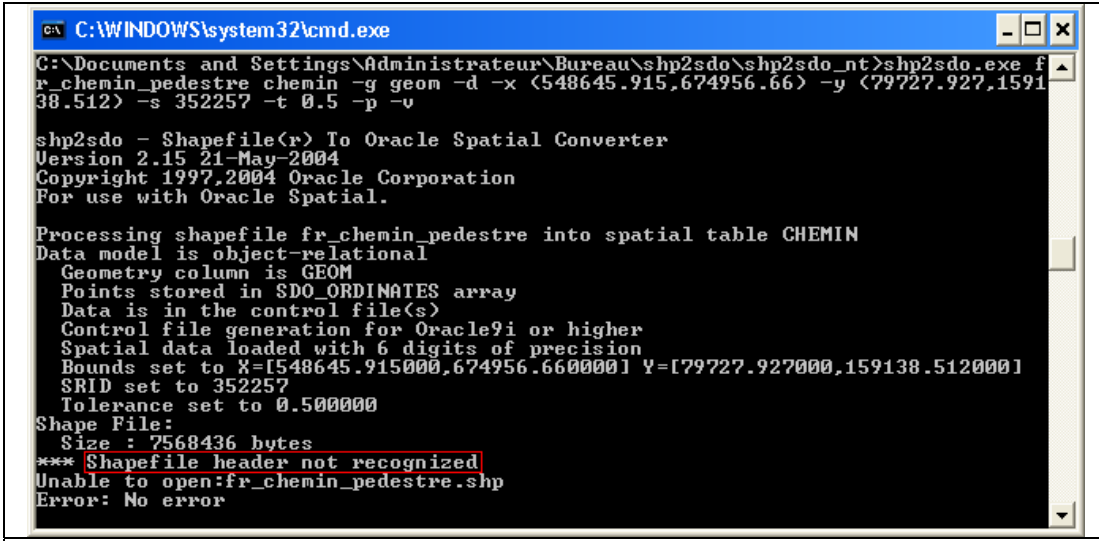
<sup>33</sup> Personne de contact : Rainer Oggier, CC Geo Chef, Service des mensurations cadastrales et de la géomatique de l'Etat du Valais

On remarque qu'une commune est représentée par un identifiant et une liste de points représentés dans le système de coordonnées nationales suisse. Aucune information sur le nom de la commune n'est signalée et une commune peut avoir jusqu'à 1500 lignes, ce qui fait 3000 coordonnées, pour délimiter son contour.

## Problèmes

Ces données posent plusieurs problèmes.

Le premier concerne l'importation des données dans une base de données Oracle. En effet, Oracle propose un utilitaire appelé « shp2sdo » permettant de convertir un fichier .shp en géométrie. Cependant, lors de l'utilisation de cet utilitaire une erreur importante est apparue :



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrateur\Bureau\shp2sdo\shp2sdo_nt>shp2sdo.exe f
r_chemin_pedestre chemin -g geom -d -x (548645.915,674956.66) -y (79727.927,1591
38.512) -s 352257 -t 0.5 -p -v

shp2sdo - Shapefile(r) To Oracle Spatial Converter
Version 2.15 21-May-2004
Copyright 1997,2004 Oracle Corporation
For use with Oracle Spatial.

Processing shapefile fr_chemin_pedestre into spatial table CHEMIN
Data model is object-relational
Geometry column is GEOM
Points stored in SDO_ORDINATES array
Data is in the control file(s)
Control file generation for Oracle9i or higher
Spatial data loaded with 6 digits of precision
Bounds set to X=[548645.915000,674956.660000] Y=[79727.927000,159138.512000]
SRID set to 352257
Tolerance set to 0.500000
Shape File:
Size : 7568436 bytes
*** Shapefile header not recognized
Unable to open: fr_chemin_pedestre.shp
Error: No error
```

Figure 67 : Erreur shp2sdo

Apparemment, les données de l'Etat doivent posséder un header non compatible avec cet utilitaire.

Deuxièmement, les données ne possèdent pas d'information sur l'altitude et de plus, elles ne sont pas très pratiques à utiliser. En effet, une commune est représentée par un numéro et non par son nom. Ainsi, il faut parcourir plus de 65000 lignes pour rechercher une commune désirée. Il est donc difficile de trouver des informations précises. De plus, les fichiers concernant les routes et les chemins sont nettement plus grands que celui des communes.

Troisièmement, SQLPlus ne permet pas d'entrer, lors de la création de géométries, plus de 999 éléments, ce qui correspond à 499 coordonnées en 2D ou 333 en 3D. En partant du principe qu'une commune peut être représentée par plus de 1500 coordonnées, il devient difficile d'être précis et de savoir quels points garder pour avoir des représentations plus ou moins correctes.

En conclusion, il n'est pas réellement imaginable qu'en 3 semaines, il soit possible de mettre en place une base de données basée sur les données des mensurations cadastrales. Dans l'état actuel des choses, les données devraient subir trop de transformation. La décision a donc été prise de laisser de côté ces données et de créer un scénario fictif permettant de montrer les différents aspects du travail.

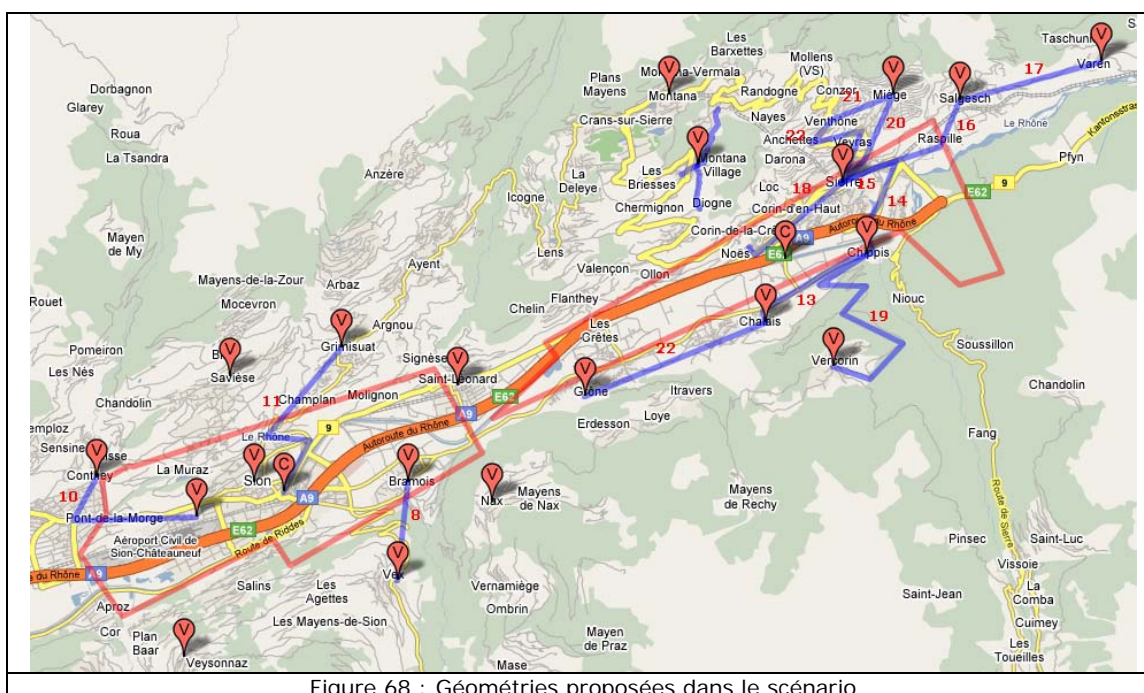
## Scénario

Le scénario mis en place propose une série de données<sup>34</sup> présentées dans l'image ci-dessous. Ces informations ont été prises pour une partie dans les données fournies par l'Etat du Valais mais la plupart ont été calculées manuellement.

Les informations stockées sont les suivantes :

- Les zones rouges représentent les communes de Sion et de Sierre.
- Les marqueurs correspondent aux villes et villages
- Les lignes bleues se rapportent aux routes utilisées pour la création d'itinéraires. Celles-ci sont numérotées

L'ensemble de ces données permet d'avoir des résultats pour tous les types de recherches d'itinéraires présentées dans le processus.



Cette image se trouve dans son format original sur le DVD, dans le dossier « Documents/Données/ ».

<sup>34</sup> Les informations entrées dans la base de données doivent suivre certains critères. Ces données doivent être vectorielles, posséder 3 dimensions (latitude, longitude et altitude) et chaque segment doit représenter un tronçon de route sans intersection.

# Gestion de projet

## Cahier des charges

---

### Contexte du projet

Ce projet est réalisé dans le cadre du travail de diplôme de la formation d'informaticien de gestion HES à la HEVs de Sierre. Ce travail se déroule sur une période de 12 semaines, du 25 septembre 2006 au 18 décembre 2006.

Une présentation du travail réalisé aura lieu courant décembre 2006.

### Présentation du projet

De plus en plus d'applications de gestion sont appelées à intégrer des données spatiales liées à du positionnement géographique ou à des itinéraires. Les domaines d'activité touchés sont très variés : gestion de l'aménagement du territoire, agriculture, prévention des risques naturels, gestion des infrastructures routières et médecine ne sont qu'une infime part ces domaines.

La centralisation des données géo-spatiales dans une base de données permet de garder une universalité et une intégrité des données ainsi que d'avoir une mise à jour fréquente de celles-ci, ce qui n'était pas envisageable dans le passé avec de simples cartes.

### Objectifs

L'objectif de ce projet est tout d'abord d'explorer les fonctionnalités GIS (Geographical Information Systems ou Système d'Informations Géographiques en français) offertes par les bases de données. Un système d'informations géographiques constitue l'ensemble du processus visant à la gestion des données géo-spatiales : acquisition, archivage, abstraction, analyse et affichage des données (5A). Le but de ce travail n'est cependant pas de gérer l'intégralité d'un GIS, mais de se concentrer plus particulièrement sur la partie analyse et l'utilisation des données et fonctions du GIS dans une application d'informatique de gestion.

Le second objectif sera donc d'analyser plus en détail la structure des données géo-spatiales et les fonctions/algorithmes existants/à développer pour exploiter ces informations.

Finalement, il faudra pouvoir gérer les informations spatiales avec les données de gestion classique.

### Démarche projetée en vue d'atteindre les objectifs

Le projet se déroulera en deux grandes phases : une phase de recherche d'information et d'analyse suivie d'une phase de développement du prototype.

#### Recherche et analyse

Le travail débutera tout d'abord par la mise en place d'une documentation relative à l'état de l'art de la représentation des données spatiales dans les SGBDR courants. Un comparatif sera établi pour permettre d'avoir une vue d'ensemble de ces solutions, tant du point de vue des bases de données que des logiciels SIG existants.

Par la suite, un document sera rédigé dans le but de connaître les normes existantes ainsi que les fonctionnalités, les rôles,... des systèmes d'informations géographiques. Une documentation globale d'un GIS sera faite mais celui-ci ne sera pas développé dans son intégralité dans le prototype.



En parallèle, une analyse poussée sera effectuée pour comprendre le fonctionnement et la gestion d'une base de données géographique. De là, découlera un dossier détaillé du travail effectué sur les analyses puis sur le prototype décrit ci-dessous.

Finalement, des recherches sur la gestion des données GPS seront faites afin d'analyser les possibilités d'intégration de ces informations dans des bases de données relationnelles.

#### Développement du prototype

La partie pratique de ce projet consistera en l'élaboration d'une application web prototype, proposant à l'utilisateur des randonnées ou des itinéraires correspondants à des critères sélectionnés :

- type de chemin
  - o Montagne
  - o Bisse
  - o Route
  - o ...
- difficulté
- dénivelé
- type d'activité
  - o VTT
  - o Vélo de route
  - o Marche
  - o ...
- Type de parcours
  - o Circulaire
  - o Droit
- ...

Ce site devra également permettre à l'utilisateur possédant un GPS d'enrichir la base de données grâce aux informations recueillies sur le terrain. Ainsi, ce site collaboratif donnera la possibilité d'avoir accès à des informations de plus en plus précises et de plus en plus nombreuses.

Actuellement, toutes les possibilités des bases de données géographiques n'étant pas connues, il n'est pas encore évident de savoir exactement à quoi correspondra le prototype. De plus, comme dit précédemment, un système d'information géographique est le processus complet arrivant à la mise en place d'un système. Il n'est pas dans l'intérêt de ce projet de tout traiter. Cependant, celui-ci devra donner la possibilité de comprendre au mieux la structure et la gestion des données spatiales ainsi que démontrer la manière de gérer des informations géographiques avec des données de gestion classique.



## Découpage du travail

---

Ce projet de douze semaines a été découpé en cinq étapes principales.

La partie « Recherche » a permis d'analyser et de se documenter sur les solutions existantes de gestion de données géographiques. Afin d'avoir une base solide sur le sujet, dix jours y ont été consacrés.

L'étape « Documentation » consistait à la création de documents expliquant et détaillant les informations trouvées dans la phase « Recherche » ainsi que sur la norme OpenGIS et les systèmes d'information géographiques.

Les divers tests et la création de la base de données ont été faits lors de l'« analyse et de la modélisation ». Les tests ont permis de choisir Oracle Spatial comme base de données.

Le « développement » a été la phase la plus longue et la plus complexe de ce projet. L'élaboration du site Internet ainsi que la mise sur pied d'un scénario sont les points principaux de cette partie.

Finalement, l'étape « Finalisation » comprenait la mise en commun de la documentation déjà produite en début de projet ainsi que la réalisation du rapport final.

Les fichiers .mpp correspondant à ces diverses étapes se trouvent sur le DVD, dans le dossier « /Documents/Planification/ »

## Planification

### Initiale

<input type="checkbox"/> <b>Projet GIS</b>	<b>59.14 jours</b>	<b>Lun 25.09.06</b>	<b>Ven 15.12.06</b>
Séance	1 jour	Lun 25.09.06	Lun 25.09.06
Planification	1 jour	Mar 26.09.06	Mar 26.09.06
Cahier des charges	1 jour	Mer 27.09.06	Mer 27.09.06
<input type="checkbox"/> <b>Recherches</b>	<b>10.5 jours</b>	<b>Jeu 28.09.06</b>	<b>Jeu 12.10.06</b>
Bases de données	3 jours	Jeu 28.09.06	Lun 02.10.06
Outils/logiciels existants	3 jours	Mar 03.10.06	Jeu 05.10.06
GIS	3 jours	Ven 06.10.06	Mar 10.10.06
GPS	1 jour	Mer 11.10.06	Mer 11.10.06
Choix de la solution	0.5 jour	Jeu 12.10.06	Jeu 12.10.06
<input type="checkbox"/> <b>Documentation</b>	<b>7 jours</b>	<b>Jeu 12.10.06</b>	<b>Lun 23.10.06</b>
Finalisation du cahier des charges	1 jour	Jeu 12.10.06	Ven 13.10.06
Comparatif et état de l'art de la représentation graphique dans les SGBDR	2 jours	Ven 13.10.06	Mar 17.10.06
GIS et OpenGIS	3 jours	Mar 17.10.06	Ven 20.10.06
GPS	1 jour	Ven 20.10.06	Lun 23.10.06
<input type="checkbox"/> <b>Analyse et modélisation</b>	<b>7 jours</b>	<b>Lun 23.10.06</b>	<b>Mer 01.11.06</b>
<input type="checkbox"/> <b>Analyse BD Exemple</b>	<b>4 jours</b>	<b>Lun 23.10.06</b>	<b>Ven 27.10.06</b>
MySQLSpatial et SpatialWare	2 jours	Lun 23.10.06	Mer 25.10.06
Oracle Spatial et Locator	2 jours	Mer 25.10.06	Ven 27.10.06
Use cases	1 jour	Ven 27.10.06	Lun 30.10.06
Modélisation de la base de données	3 jours	Ven 27.10.06	Mer 01.11.06
<input type="checkbox"/> <b>Développement</b>	<b>25 jours</b>	<b>Mer 01.11.06</b>	<b>Mer 06.12.06</b>
Prototypes écran	1 jour	Mer 01.11.06	Jeu 02.11.06
Base de données	8 jours	Jeu 02.11.06	Mar 14.11.06
Fonctions et algorithmes	8 jours	Mar 14.11.06	Ven 24.11.06
Application	5 jours	Ven 24.11.06	Ven 01.12.06
Documentation détaillée du prototype	3 jours	Ven 01.12.06	Mer 06.12.06
<input type="checkbox"/> <b>Finalisation</b>	<b>6 jours</b>	<b>Mer 06.12.06</b>	<b>Jeu 14.12.06</b>
Documentation	4 jours	Mer 06.12.06	Mar 12.12.06
Préparation de la présentation	2 jours	Mar 12.12.06	Jeu 14.12.06
Fin du projet	0 jour	Jeu 14.12.06	Jeu 14.12.06
<input type="checkbox"/> <b>Séance</b>	<b>55.14 jours</b>	<b>Ven 29.09.06</b>	<b>Ven 15.12.06</b>

Figure 69 : Planification initiale

### Modification

La planification initiale a subi quelques modifications en cours de projet suite à des problèmes apparus lors du développement des fonctions et des algorithmes.

L'incidence de ces problèmes sur la planification a été le rallongement de la tâche « Fonctions et algorithmes ». Afin de terminer le projet à temps, il a donc fallu supprimer quelques jours sur la tâche « Base de données », qui a été par la suite renommée « Intégration des données ».

Ces modifications ainsi que le retard dans l'acquisition des données, ont eu comme conséquence l'impossibilité d'utiliser les données fournies par l'Etat du Valais, celles-ci demandant beaucoup de travail de transformation. Le scénario mis en place au final, n'a nécessité que trois jours et non huit comme prévu initialement.

De plus, la rédaction de la documentation ayant été faite à mesure, il a été possible de diminuer le nombre de jours nécessaire à la documentation du prototype.

Voici la nouvelle planification :

<b>Projet GIS</b>	<b>59.5 jours</b>	<b>Lun 25.09.06</b>	<b>Ven 15.12.06</b>
Séance	1 jour	Lun 25.09.06	Lun 25.09.06
Planification	1 jour	Mar 26.09.06	Mar 26.09.06
Cahier des charges	1 jour	Mer 27.09.06	Mer 27.09.06
<b>Recherches</b>	<b>10 jours</b>	<b>Jeu 28.09.06</b>	<b>Mer 11.10.06</b>
Bases de données	3 jours	Jeu 28.09.06	Lun 02.10.06
Outils/logiciels existants	3 jours	Mar 03.10.06	Jeu 05.10.06
GIS	3 jours	Ven 06.10.06	Mar 10.10.06
GPS	1 jour	Mer 11.10.06	Mer 11.10.06
<b>Documentation</b>	<b>7 jours</b>	<b>Jeu 12.10.06</b>	<b>Ven 20.10.06</b>
Finalisation du cahier des charges	1 jour	Jeu 12.10.06	Jeu 12.10.06
Comparatif et état de l'art de la représentation graphique dans les SGBDR	2 jours	Ven 13.10.06	Lun 16.10.06
GIS et OpenGIS	3 jours	Mar 17.10.06	Jeu 19.10.06
GPS	1 jour	Ven 20.10.06	Ven 20.10.06
<b>Analyse et modélisation</b>	<b>8.5 jours</b>	<b>Lun 23.10.06</b>	<b>Jeu 02.11.06</b>
<b>Analyse BD Exemple</b>	<b>4.5 jours</b>	<b>Lun 23.10.06</b>	<b>Ven 27.10.06</b>
MsSQLSpatial et SpatialWare	2 jours	Lun 23.10.06	Mar 24.10.06
Oracle Spatial et Locator	2 jours	Mer 25.10.06	Jeu 26.10.06
Choix de la solution	0.5 jour	Ven 27.10.06	Ven 27.10.06
Use cases	1 jour	Ven 27.10.06	Lun 30.10.06
Modélisation de la base de données et création de la base de données	3 jours	Lun 30.10.06	Jeu 02.11.06
<b>Développement</b>	<b>25 jours</b>	<b>Jeu 02.11.06</b>	<b>Jeu 07.12.06</b>
Fonctions et algorithmes	14 jours	Jeu 02.11.06	Mer 22.11.06
Application	5 jours	Mer 22.11.06	Mer 29.11.06
Intégration des données	3 jours	Mer 29.11.06	Lun 04.12.06
Prototypes écran	1 jour	Lun 04.12.06	Mar 05.12.06
Documentation détaillée du prototype et de la création de la VMWare	2 jours	Mar 05.12.06	Jeu 07.12.06
<b>Finalisation</b>	<b>6 jours</b>	<b>Jeu 07.12.06</b>	<b>Ven 15.12.06</b>
Documentation	5 jours	Jeu 07.12.06	Jeu 14.12.06
Préparation de la présentation	1 jour	Jeu 14.12.06	Ven 15.12.06
Fin du projet	0 jour	Ven 15.12.06	Ven 15.12.06

Figure 70 : Planification modifiée

Au final, toutes ces modifications n'ont pas eu d'influences sur la durée du projet.

Statistiques sur le projet pour '03_Planification Finale.mpp'			
	Début		Fin
En cours	Lun 25.09.06		Ven 15.12.06
Planifié	Lun 25.09.06		Ven 15.12.06
Réel	Lun 25.09.06		Ven 15.12.06
Variation	0j		0j
	Durée	Travail	Coût
En cours	59.5j	395.5h	CHF 31'640.00
Planifié	59.5j	395.5h	CHF 31'640.00
Réel	59.5j	395.5h	CHF 31'640.00
Restant	0j	0h	CHF 0.00
% achevé :			
Durée :	100%	Travail :	100%
			Fermer

Figure 71 : Statistiques du projet

## Déroulement du projet

---

Un total de plus de 590 heures a été nécessaire à l'élaboration de ce travail.

### *Semaine 1-2*

Les deux premières semaines ont surtout été une période de démarrage de projet. En effet, plusieurs séances ont été organisées et de nombreuses recherches ont permis de se familiariser avec les bases de données et outils existants.

Durant ces deux semaines, la planification et la rédaction du cahier des charges ont également été effectuées.

Total des heures : 83 h

### *Semaine 3*

Afin de pouvoir tester quelques outils existants, Oracle 9i a été installé. Cela a demandé des recherches supplémentaires sur l'utilisation d'Oracle Spatial. De plus, des informations sur les normes existantes, sur la gestion des données GPS et sur les systèmes d'information géographique ont été examinées.

Total des heures : 54 h

### *Semaine 4*

La quatrième semaine a servi à finaliser une partie de la documentation élaborée lors de la semaine 3 ainsi qu'à mettre sur pied une première version des use cases et du modèle physique de données.

De plus, des tests plus poussés sur Oracle Spatial ont été effectués.

Total des heures : 48 h

### *Semaine 5*

La semaine 5 correspond à la création d'une base de données exemple ainsi qu'au début du développement des fonctions et procédures nécessaires à la recherche dynamique d'itinéraires.

Total des heures : 50 h

### *Semaine 6*

L'élaboration du document central de ce projet, le « processus de recherche d'itinéraires », a été débuté lors de la semaine 6. De plus, un rendez-vous avec monsieur Serge Gaudin a été pris afin de trouver des données réelles à intégrer à la base de données.

Les tâches commencées en semaine 6 ont été poursuivies.

Total des heures : 41 h

### *Semaine 7*

La semaine 7 a été une période de grande réflexion. Il a fallu réfléchir au meilleur moyen de gérer les données afin de trouver des itinéraires de façon dynamique. En définitif, la décision a été d'utiliser des Graphes.

Serge Gaudin n'ayant pas pu me renseigner de manière précise sur les données de l'Etat du Valais, un rendez-vous a été programmé avec monsieur Yves Molk du service des routes et cours d'eau de l'Etat du Valais et monsieur Pierre Gonin d'Inser. Les données StradaDB ne correspondant pas au besoin pour ce projet, un contact avec

monsieur Rainer Oggier, chef de la section géomatique de l'Etat du Valais, gérant l'ensemble des données géographiques, a été pris.

Total des heures : 52 h

#### *Semaine 8-9-10*

Les trois semaines suivantes ont été entièrement consacrées au développement de l'application web (Graphes, procédures et fonctions Oracle, site Internet, design,...). Lors de la semaine 9, les données de l'Etat du Valais ont été reçues mais le temps étant trop restreint pour les transformer selon les besoins de l'application, elles ont été mises de côté. En contrepartie, un scénario fictif mais représentant convenablement la réalité a été mis en place.

Total des heures : 155 h

#### *Semaine 11-12*

Le début de la semaine 11 a permis de corriger les derniers problèmes de l'application web et de créer une image VMWare contenant l'ensemble de la partie pratique de ce projet.

Le temps restant a été consacré à la finalisation de la documentation et à la préparation des supports à joindre au dossier final.

Total des heures : 110 h

## Synthèse

De nos jours, il devient de plus en plus intéressant d'intégrer des données spatiales à des applications de gestion courantes. Cela s'explique par le fait que l'analyse des données classiques est grandement enrichie lorsque celles-ci peuvent être rattachées à leur contexte géographique.

Ce projet a permis d'apprendre à traiter ces données spatiales et d'avoir une bonne vision d'un système d'information géographique.

De plus, il a été possible de se familiariser avec de nouveaux logiciels et outils tels qu'Oracle Spatial, ChartFX, ou l'API GoogleMaps.

D'autre part, ce projet a nécessité un travail de réflexion afin de palier aux problèmes d'adaptation des données spatiales aux divers logiciels utilisés, comme par exemple le fait que Visual Studio n'accepte pas les géométries Oracle Spatial ou encore qu'aucune base de données ne gère réellement bien la troisième dimension (altitude).

En définitive, les objectifs définis en début de projet ont été atteints puisque qu'une application gérant des données géographiques avec des données classiques a été développée.

La principale force de ce travail est l'élaboration d'un site innovant, pouvant amener un attrait supplémentaire aux touristes voulant visiter le Valais ou toute autre région du monde. En effet, ce site propose une recherche d'itinéraires de randonnées de façon dynamique et selon différents critères de sélection, alors que jusqu'à maintenant seulement des sites proposant des parcours fixes existaient.

De plus, l'affichage sur GoogleMaps du trajet, la représentation graphique du dénivelé ainsi que la possibilité d'exportation vers un format GPS sont des plus-values qu'un utilisateur du site saura apprécier.

Finalement, le dernier point positif est que ce site peut facilement être adapté à n'importe quel endroit dans le monde.

Suite à un manque de temps, il n'a pas été possible de développer l'intégralité du site web. Les fonctionnalités d'ajout et de modification des routes ont été pensées mais pas mises en place.

De plus, les données de l'Etat du Valais n'étant pas exploitables sans transformation, il n'a pas été possible d'intégrer des données réelles dans la base de données. Cela aurait permis d'avoir une meilleure idée de la gestion d'un grand nombre de données géographiques.

Il serait donc envisageable dans un futur proche de développer l'ensemble du site pour le rendre collaboratif, ainsi que d'ajouter de nouveaux formats d'exportation pour les itinéraires de randonnées.

De plus, il pourrait être intéressant de lier cette application avec des applications mobiles, puisque de plus en plus d'appareils portables intègrent une fonction GPS.

## Remerciements

J'aimerais remercier toutes les personnes qui ont contribué à l'élaboration de mon projet :

Monsieur Yann Bocchi pour sa disponibilité et ses précieux conseils. Son enthousiasme pour le sujet a créé un climat agréable et m'a permis de faire face aux difficultés rencontrées.

Tous mes remerciements vont également à Serge Gaudin, Yves Molk et Pierre Gonin pour le temps qu'ils ont consacré à m'expliquer la façon dont StradaDB gère les routes valaisannes. Ils ont été à l'écoute de mes besoins et m'ont orienté vers les personnes adéquates.

Monsieur Rainer Oggier pour m'avoir fourni les données de l'Etat du Valais.

Finalement, merci à toutes les personnes qui m'ont supporté tout au long de ce travail.

## Sources

### Bibliographie

---

Les documents cités dans cette bibliographie se trouvent sur le DVD dans le répertoire  
« Document/Sources externes/ »

### Comparatif

#### *Oracle*

Oracle Spatial User's Guide and Reference 10g Release 2.pdf

<http://www.oracle.com/technology/documentation/spatial.html>

#### *SQL Server*

SpatialWare User Guide.pdf

<http://extranet.mapinfo.com/support/documentation/manuals.cfm>

#### *MapInfo*

MapInfo Professional.pdf

<http://www.i2g.fr/pdf/mapinfo.pdf>

#### *Oracle Application Server MapViewer*

Oracle Application Server – MapViewer user's guide 10g release 2.pdf

[http://www.oracle.com/technology/products/mapviewer/mapviewer\\_doc\\_index.html](http://www.oracle.com/technology/products/mapviewer/mapviewer_doc_index.html)

#### *Quantum GIS*

Quantum GIS User guide.pdf

[http://qgis.org/index.php?option=com\\_content&task=view&id=106&Itemid=65](http://qgis.org/index.php?option=com_content&task=view&id=106&Itemid=65)

### Norme

OpenGis SFS for SQL Rev 1.1.pdf

<http://www.opengeospatial.org/standards/sfs>

### GIS – SIG

Qu'est-ce qu'un SIG.pdf

Laboratoire de cartographie appliquée - Elisabeth HABERT - Institut de recherche pour le développement

<http://www.cartographie.ird.fr/publi/documents/sig1.pdf>

Mise en œuvre d'un SIG.pdf

Institut Atlantique d'aménagement des territoires

<http://www.iaat.org/telechargement/sig.pdf>

Intro SIG.pdf

Institut National Agronomique Paris-Grignon

<http://www.forumsig.org/>

SIG Notions fondamentales.ppt

Abdelali Taouss - Agence Nationale de la Conservation Foncière du cadastre et de la Cartographie

<http://www.forumsig.org/>



## GPS

Schéma de définition d'un fichier GPX

Gpx.xsd

<http://www.topografix.com/GPX/1/1/gpx.xsd>

## Intégration des données

StradaDB

Normalisation.pdf

<http://topo.epfl.ch/siroute/syrrou/documents.html>

## Oracle Spatial

Création des géométries

OracleSpatial.pdf

<http://lbdwww.epfl.ch/e/teaching/SlidesBDA/OracleSpatial.pdf>

## ChartFX

ChartFX Quick start guide.pdf

<http://eu.softwarefx.com/sfxNetProducts/cfxVs2005/>

## URL

---

### Comparatif

#### *Oracle*

Comparison of the Oracle Spatial and Locator features  
<http://www.mattpopper.com/?p=26>

#### *SQLServer*

Site officiel de MsSQLSpatial  
<http://www.codeplex.com/Wiki/View.aspx?ProjectName=MsSqlSpatial>

Download de SpatialWare  
<http://extranet.mapinfo.com/products/Download.cfm?ProductID=1141&productcategoryid=>

#### *MySql*

Documentation sur les données spatiales dans MySQL  
<http://forum.mysql.com/doc/refman/5.0/fr/spatial-extensions.html>

#### *PostGIS*

Documentation PostGIS  
<http://postgis.refractions.net/docs/>

#### *DB2*

Information sur le produit DB2 Spatial Extender  
[www-306.ibm.com/software/info/ecatalog/fr\\_FR/products/C105970D58974G08.html](http://www-306.ibm.com/software/info/ecatalog/fr_FR/products/C105970D58974G08.html)

#### *ArcGIS*

Informations sur ArcGIS 9  
<http://www.esrifrance.fr/produit/arcgis.asp>  
ArcView  
<http://www.esri.com/software/arcgis/arcview/about/features.html>

#### *MapInfo*

Site officiel de MapInfo  
<http://www.mapinfo.com>

#### *SharpMap*

Site officiel de SharpMap  
<http://www.codeplex.com/Wiki/View.aspx?ProjectName=SharpMap>

### Norme

Wikipedia  
[www.wikipedia.org](http://www.wikipedia.org)

Open Geospatial Consortium  
<http://www.opengeospatial.com/>

Informations sur la norme OpenGIS  
<http://eden.ign.fr/std/>

## GPS

Wikipedia

[www.wikipedia.org](http://www.wikipedia.org)

Site officiel du format GPX

<http://www.topografix.com>

Format PCX5

<http://home.online.no/~sigurdhu/MapSource-text.htm>

Format NMEA 0183

<http://www.kh-gps.de/nmea-faq.htm>

Information sur les GPS

[http://www.ac-orleans-tours.fr/svt/SIG/dossiers/sig\\_local/HTML/page3b.htm](http://www.ac-orleans-tours.fr/svt/SIG/dossiers/sig_local/HTML/page3b.htm)

Information sur les GPS

[http://www.skitour.fr/articles/read\\_78.html](http://www.skitour.fr/articles/read_78.html)

## API Google Maps

Site officiel de GoogleMaps API

<http://www.google.com/apis/maps>

## PL/SQL

Théorie sur le PL/SQL

<http://infolab.stanford.edu/~ullman/fcdb/oracle/or-plsql.html>

Exemple PS et cursor

<http://forums.microsoft.com/MSDN-FR/ShowPost.aspx?PostID=711549&SiteID=12>

<http://support.microsoft.com/kb/321718/>

<http://www.zdnet.fr/livres-blancs/0,39035134,60246730p-39000727q,00.htm>

[http://www.zdnet.fr/builder/programmation/windows\\_dotnet/0,39021036,213314,00.htm](http://www.zdnet.fr/builder/programmation/windows_dotnet/0,39021036,213314,00.htm)

[http://www.infini-fr.com/Sciences/Informatique/Base\\_de\\_donnees/Relationnel/Oracle/plsql.html](http://www.infini-fr.com/Sciences/Informatique/Base_de_donnees/Relationnel/Oracle/plsql.html)

## Oracle Spatial

Oracle by exemple – Comment utiliser Oracle Spatial pour effectuer des analyses spatiales

<http://www.oracle.com/technology/obe/obe9ir2/obe-cnt/spatial/spatial.htm>

Systèmes de coordonnées

[http://www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96630/sdo\\_cs\\_concepts.htm](http://www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96630/sdo_cs_concepts.htm)

<http://topo.epfl.ch/documents/transcoco/projch/projch.php>

sdo\_tolerance et arc\_tolerance

[http://www.cs.uvm.edu/oracle9doc/appdev.901/a88805/sdo\\_intr.htm#884590](http://www.cs.uvm.edu/oracle9doc/appdev.901/a88805/sdo_intr.htm#884590)

[http://dba-services.berkeley.edu/docs/oracle/manual-10gR2/appdev.102/b14255/sdo\\_objgeom.htm#i867636](http://dba-services.berkeley.edu/docs/oracle/manual-10gR2/appdev.102/b14255/sdo_objgeom.htm#i867636)

Index spatiaux

[http://www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo\\_exten.htm](http://www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_exten.htm)

Récupération des coordonnées

<http://forums.oracle.com/forums/thread.jspa?messageID=427912&#427912>

## Outils

---

Durant ce projet, différents outils ont été nécessaires. Tous ces utilitaires se trouvent sur le DVD dans le dossier « Outils »

### WebApplication

L'installation de base de Visual Studio 2005 ne permet pas de créer une application web. C'est pour cela qu'il faut installer en plus Visual Studio 2005 Web Application Projects téléchargeable gratuitement à l'adresse suivante : <http://msdn2.microsoft.com/en-us/asp.net/aa336618.aspx>

Pour pouvoir employer cet utilitaire, il faut également installer une mise à jour permettant à Visual Studio 2005 de supporter les applications web : <http://www.microsoft.com/downloads/details.aspx?FamilyId=8B05EE00-9554-4733-8725-3CA89DD9BFCA&displaylang=en>

### ShapeViewer

ShapeViewer est un petit utilitaire gratuit permettant de visualiser graphiquement les fichiers shapefile (.shp).

Il est possible de le télécharger sur <http://www.shapeviewer.com/>

### Shp2Sdo

Shp2sdo est un utilitaire fourni par Oracle permettant de convertir des fichiers shapefile en géométrie Oracle.

<http://www.oracle.com/technology/software/products/spatial/htdocs/xplatformsoft.html>

## ChartFX<sup>35</sup>

ChartFX est un outil développé par la société Software FX. Il se décline en plusieurs versions dont une s'adaptant spécialement à Visual Studio 2005. ChartFx permet d'afficher des graphiques facilement personnalisables.

Après avoir installé le produit, un nouveau composant s'ajoute dans le toolbox.

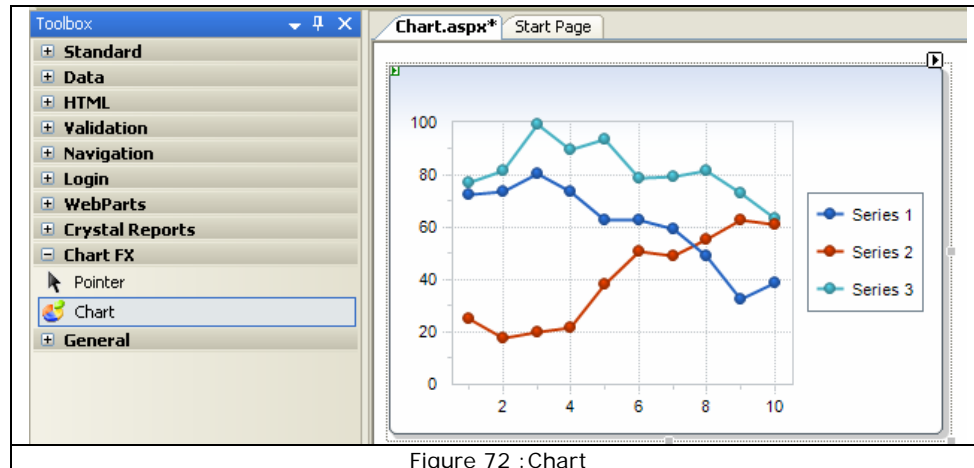


Figure 72 : Chart

Il est ensuite possible de personnaliser le graphe sur divers aspects :

- Type de graphe :
  - o Lignes et courbes
  - o Barres
  - o Pyramides
  - o ...
- Thème de couleur du graphe
- Axes et cadrillage du graphe
- Légende, titre, points
- Affichage de DataGrid et d'une Toolbar
- ...

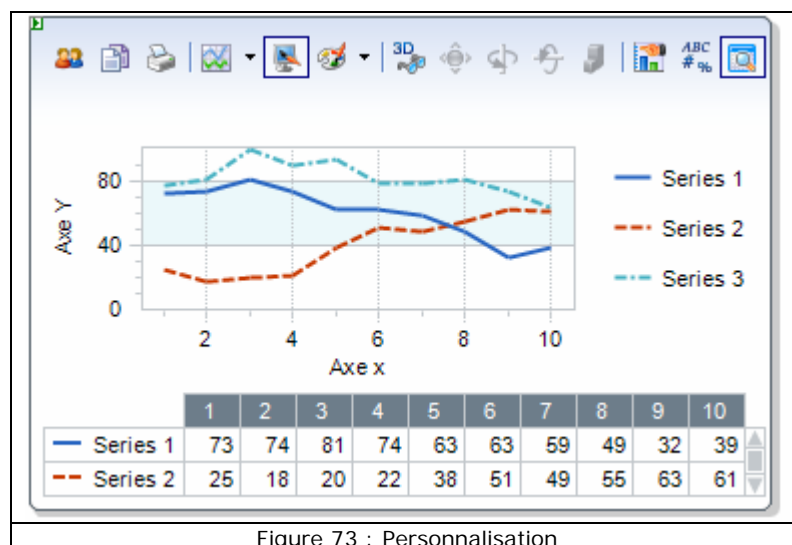


Figure 73 : Personnalisation

<sup>35</sup> <http://eu.softwarefx.com/sfxNetProducts/cfxVs2005>

De plus, le « Sample and resource center » installé automatiquement avec ChartFx, propose un guide et des exemples détaillés pour apprendre à utiliser ce composant.

Le prix de ce produit est de 2699 \$ pour la version WebForms et de 1599 \$ pour la version WinForms.

### Framework .Net 2.0

Pour pouvoir installer le site web, il est nécessaire de posséder le Framework .Net 2.0. Il est téléchargeable gratuitement sur le site de Microsoft à l'adresse suivante :

<http://www.microsoft.com/downloads/details.aspx?displaylang=fr&FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5>

## Annexes

### Use cases

---

Ces cas d'utilisation présentent les diverses fonctionnalités à implémenter pour le site web. Cependant, lors de ce projet, toutes n'ont pas pu être développées faute de temps. Il a donc été décidé de s'occuper des fonctionnalités principales : la recherche d'itinéraires, la visualisation des informations relatives à ces itinéraires, l'affichage sur carte du trajet et l'exportation des parcours sous format GPS.

### Acteurs

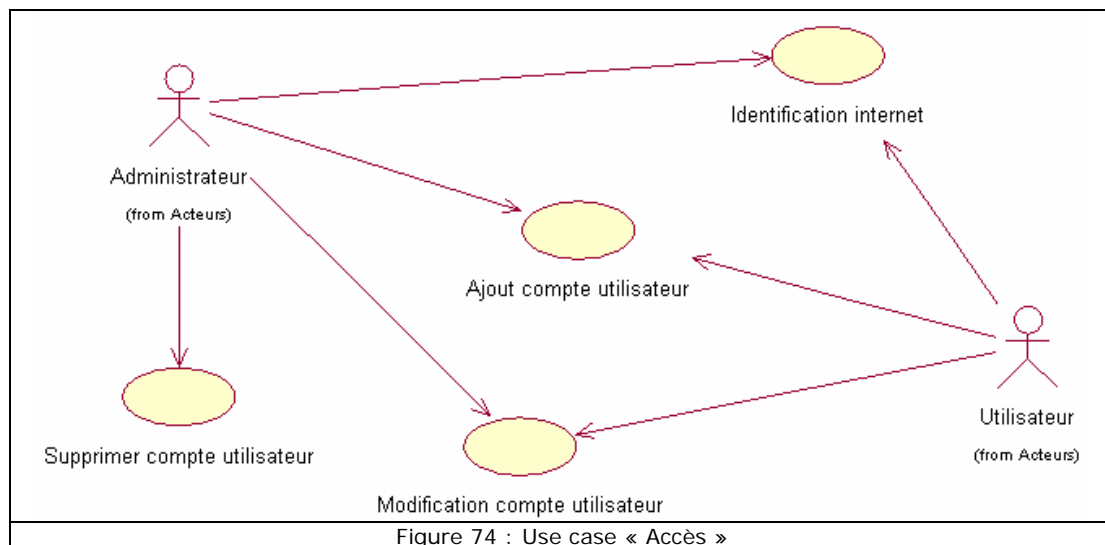
#### Utilisateur

L'utilisateur du site Internet peut visionner des itinéraires ainsi qu'ajouter et modifier des routes.

#### Administrateur

L'administrateur possède tous les droits de l'utilisateur ainsi que celui de visualiser et de valider l'ajout et la modification de nouvelles routes.

### Accès



#### Identification Internet

Tout utilisateur ou administrateur du site Internet peut accéder à son compte en s'identifiant. Il devra pour se faire entrer un nom d'utilisateur et un mot de passe. Pour un utilisateur, l'identification sur le site est nécessaire seulement pour l'ajout ou la modification de tronçons. L'administrateur devra aussi s'identifier pour visionner et valider de nouvelles routes.

### *Ajout compte utilisateur*

Un utilisateur ou un administrateur peut créer un nouveau compte. Les informations suivantes devront être fournies :

- nom
- prénom
- nom d'utilisateur
- mot de passe
- adresse e-mail

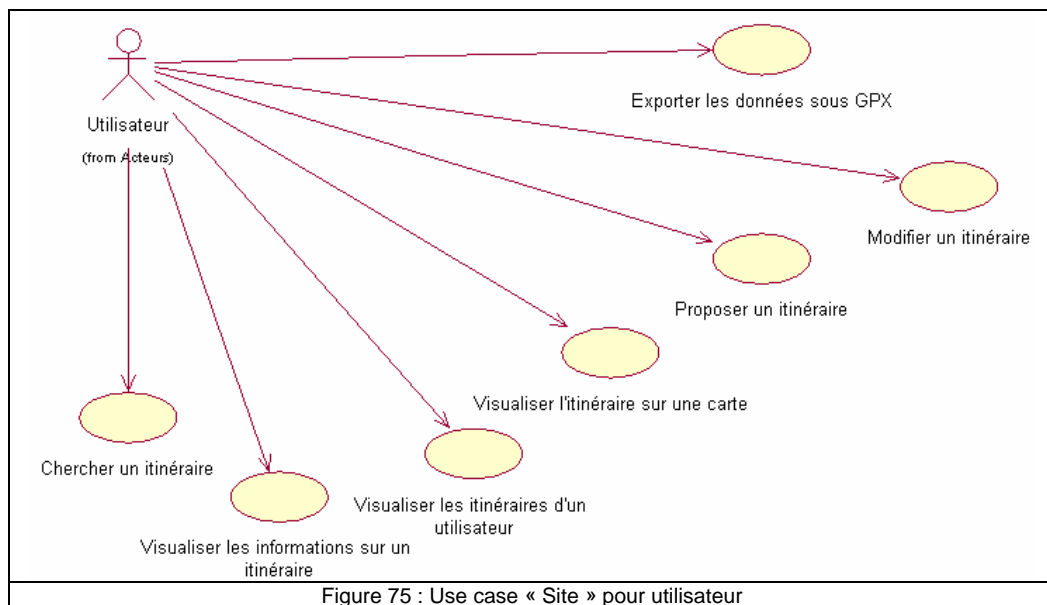
### *Modification compte utilisateur*

Un utilisateur ne peut modifier que son propre compte. L'administrateur, en revanche, a la possibilité de modifier n'importe quel compte présent sur le système.

### *Suppression compte utilisateur*

Un administrateur a le droit de supprimer un compte utilisateur.

## Fonctionnalités du site



### *Chercher un itinéraire*

Un utilisateur peut rechercher un itinéraire selon plusieurs critères :

- région, lieu
- rayon
- type d'activité (marche, vélo, ...)
- type de chemin (montagne, bisse, route,...)
- type de parcours (boucle, allé simple)
- longueur du trajet en mètre



#### *Visualiser les informations sur un itinéraire*

Après avoir fait une recherche d'itinéraires, l'utilisateur peut afficher des détails sur ce trajet. Il trouvera les informations suivantes :

- région ou lieu
- type d'activité
- type de chemin
- type de parcours
- difficulté
- dénivelé positif et négatif
- représentation graphique du dénivelé
- longueur du trajet (mètres et heures)

#### *Visualiser l'itinéraire sur une carte*

L'utilisateur peut afficher la carte relative à l'itinéraire choisi. Le chemin sera surligné sur la carte et les différents points d'intérêt présents sur le parcours seront représentés par des marqueurs.

#### *Visualiser les itinéraires d'un utilisateur*

Un utilisateur a la possibilité d'afficher la liste de toutes les routes créées par un autre utilisateur.

#### *Proposer un itinéraire*

Un utilisateur peut entrer ses données GPS dans la base de données pour permettre d'avoir de nouveaux itinéraires à proposer. Pour cela, il est obligé d'avoir créé un compte et de s'être identifié.

Les informations suivantes devront être fournies :

- fichier GPX
- type de chemin
- points d'intérêt (positionnement et informations)

Pour qu'une route soit disponible sur le site, il faudra tout d'abord qu'elle soit validée par l'administrateur.

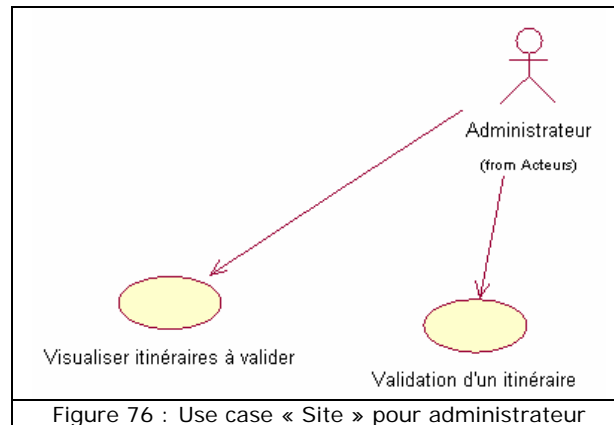
#### *Modifier un itinéraire*

Un utilisateur peut modifier une route déjà existante s'il juge que ses données sont plus précises que celles déjà présentes dans la base de données. Les mêmes informations que lors de la création d'un trajet seront demandées. Cependant, un champ supplémentaire sera présent pour permettre d'y écrire les raisons de modification du tracé.

L'administrateur pourra choisir si oui ou non, il valide la modification des données.

#### *Exporter les données sous GPX*

L'utilisateur peut exporter un itinéraire choisi sous format GPX de manière à l'intégrer dans son GPS.



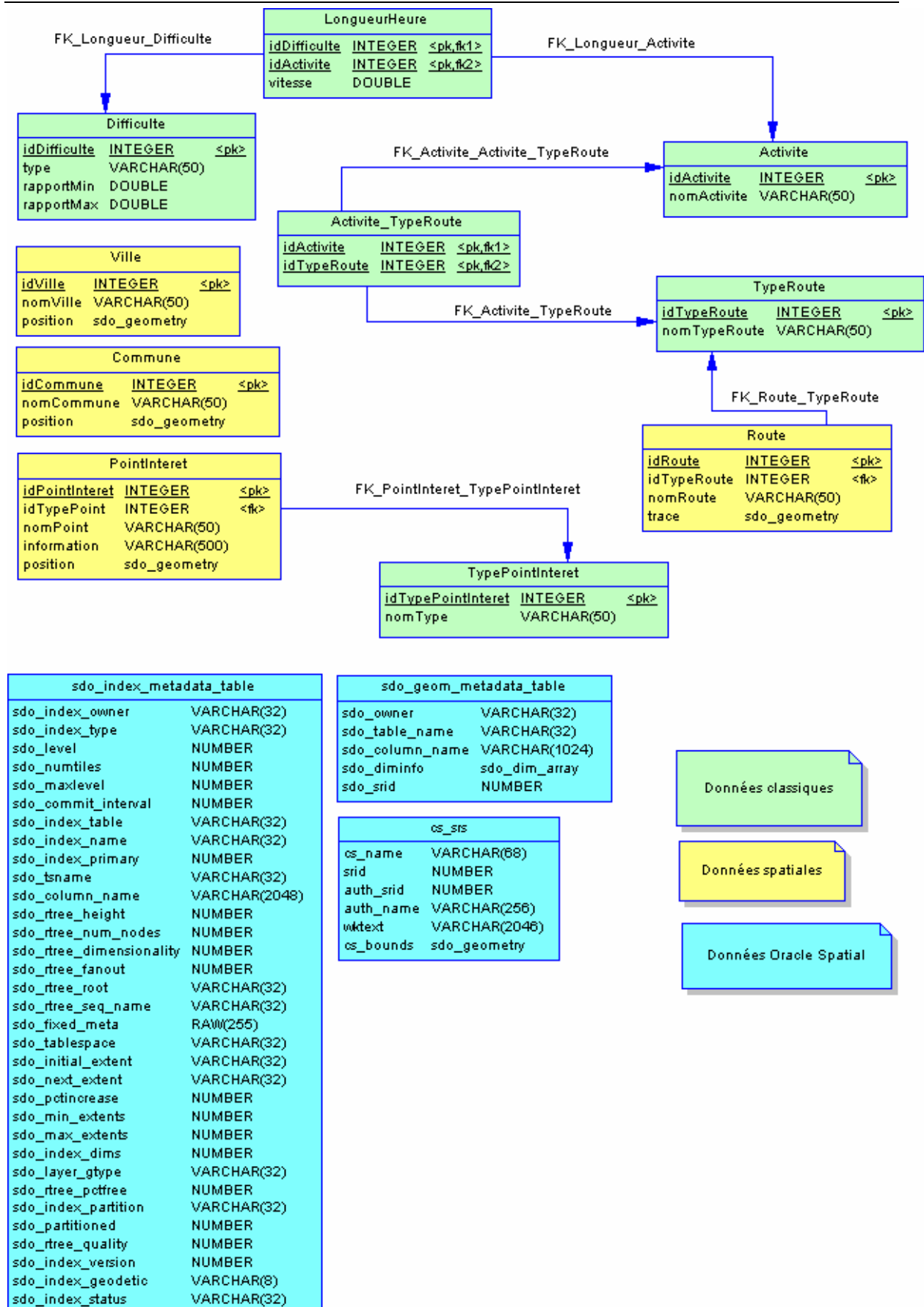
#### *Visualiser itinéraires à valider*

L'administrateur peut visualiser la liste des routes proposées par les utilisateurs après s'être identifié sur le site.

#### *Validation d'un itinéraire*

L'administrateur s'occupe de la validation lors de l'ajout ou de la modification d'un nouveau chemin. Il est envisageable que certaines routes ne soient pas acceptées.

## Modèle physique de données



Le modèle physique de données ci-dessus présente la structure de la base de données. On remarque que celle-ci est divisée en 3 parties (selon les couleurs des tables) :

- Tables représentant les données spatiales (jaune)
- Tables contenant les données de gestion classiques (vert)
- Tables du schéma mdsys présentes dans Oracle Spatial (bleu)

Ce descriptif expliquera sommairement le contenu de chaque table selon sa catégorie.

## Tables de données classiques

### Table Difficulte

Cette table permet d'analyser la difficulté d'un parcours. Grâce à un algorithme, on pourra calculer un rapport entre la longueur du trajet et son dénivelé. Il suffira de venir ensuite regarder dans cette table à quel type de difficulté correspond ce rapport.

Exemple :

<b>Difficulte</b>	<u>IdDifficulte</u> Integer	type Varchar(25)	rapportMin Double	rapportMax Double
	1	Facile	0	34
	2	Moyen	34	66
	3	Difficile	66	100

### Table LongueurHeure et Activite

Ces deux tables vont permettre de gérer la durée d'un trajet selon le type d'activité et la difficulté. Pour chaque type d'activité on va définir une vitesse moyenne selon la difficulté.

Exemple :

<b>LongueurHeure</b>	<u>IdActivite</u> Integer	<u>IdDifficulte</u> Integer	Vitesse (km/h) Double
	1	1	5,0
	1	2	4,1
	1	3	3,2
	2	1	10,5

<b>Activite</b>	<u>IdActivite</u> Integer	NomActivite Varchar(25)
	1	Marche
	2	VTT

#### *Table TypePointInteret*

Cette table contient la liste des types de points d'intérêts.

#### *Tables TypeRoute et Activite\_TypeRoute*

La table TypeRoute permet de définir pour chaque route un type. Ainsi, il sera possible de savoir si une route est goudronnée, en terre ou recouverte de cailloux.

Puis chaque type de route sera lié à une ou plusieurs activités par la table Activite\_TypeRoute.

Ainsi, après avoir choisi un type d'activité, il sera proposé à l'utilisateur de choisir le type de route qu'il désire emprunter.

Un randonneur à pied pourra choisir entre tous les types de route alors qu'un cycliste sur vélo de route ne pourra pas aller sur des routes caillouteuses.

### Tables de données spatiales

#### *Table Ville et Commune*

Ces tables stockent une liste de villes ou de communes. Les informations contenues dans ces tables sont le nom et le positionnement géographique de chaque donnée. Une ville sera représentée par un point alors qu'une commune par un polygone.

#### *Table Route*

La table route contient toutes les routes du système. Chaque route possède un nom (non obligatoire) et sa trace (géométrie). Cette table possède également un lien à la table classique TypeRoute qui permet de savoir à quel type de route celle-ci appartient.

#### *Table PointInteret*

Cette table permet de garder en mémoire une liste de lieux intéressants pour les randonneurs. Ces points pourraient être une fontaine d'eau potable, un restaurant, un hôtel,... . Les points d'intérêt seront, lors du choix d'un trajet par un utilisateur, signalés sur la carte.

### Tables mdsys

Les tables présentes dans le schéma mdsys permettent de stocker différentes informations liées aux géométries. Ce modèle évoque les principales tables de ce schéma :

- sdo\_geom\_metadata\_table : métadonnées sur chaque colonne géométrique
- sdo\_index\_metadata\_table : stockage des indexes spatiaux
- cs\_srs : stockage des systèmes de références spatiales

Il est possible de mettre à jour ces tables par l'intermédiaire de vue telle que :

- user\_sdo\_geom\_metadata
- user\_sdo\_index\_metadata

## Mise en place d'une image VMWare

---

Une image VMWare a été mise sur pied afin d'avoir accès directement et sans configuration à l'intégralité de la partie pratique de ce projet. Ainsi, après avoir démarré l'image virtuelle, il suffit de cliquer sur l'icône présente sur le bureau pour ouvrir le site web.

Ce chapitre présente les différentes étapes d'installation et de configuration nécessaires à l'élaboration de cette image. Celle-ci est construite sur un système d'exploitation Windows XP SP2. L'image se trouve sur le DVD « Image VMWare ».

### Pré requis

- IIS
- .Net Framework 2.0 (l'exécutable dotnetfx.exe présent sur le DVD dans le répertoire « Outils » permet d'installer ce framework)

## Installation du LoopBack Adaptor <sup>36</sup>

Le LoopBack Adaptor doit être installé pour éviter une erreur de connexion DHCP lors de l'installation de la base de données Oracle.

### *Processus d'installation*

- 1° Aller dans le panneau de configuration puis sur « Ajout de nouveau matériel »
- 2° Cliquer sur « oui, j'ai déjà connecté le matériel »
- 3° Au fond de la liste du matériel installé, sélectionner « Ajouter un nouveau périphérique matériel »
- 4° Choisir « Installer le matériel que je sélectionne manuellement dans la liste »
- 5° Choisir « Cartes réseaux » comme matériels communs
- 6° Finalement, choisir « Microsoft » comme fabricant ainsi que « Carte de bouclage Microsoft » comme carte réseau.
- 7° Il est également conseillé de modifier l'adresse IP de cette carte réseau. Pour cette image, l'adresse IP choisie est 192.168.26.10

## Installation d'Oracle

Ce projet nécessite l'installation de la base de données Oracle ainsi que du client Oracle. Oracle Client est obligatoire pour pouvoir accéder à la base de données depuis le site web (sur IIS)

Oracle 9i release 2 est disponible en téléchargement à l'adresse suivante : <http://www.oracle.com/technology/software/products/oracle9i/index.html> ainsi que sur le DVD dans le répertoire « Outils »

---

<sup>36</sup> <http://support.microsoft.com/kb/839013>

### *Oracle DataBase*

- 1° Produit à installer : Oracle9i Database
- 2° Type d'installation : Entreprise Edition
- 3° Base de données : Universel
- 4° Identification de la BD : nom global de BD : solinath
- 5° Jeu de caractères par défaut
- 6° Définition du mot de passe pour les utilisateurs « sys » et « system ». Pour ce projet, le mot de passe « hevs07 » a été choisi.

### *Oracle Client*

- 1° Produit à installer : Oracle9i Client
- 2° Type d'installation : Runtime

## Configuration d'Oracle

Cette partie du document explique comment configurer Oracle afin de pouvoir accéder au site Internet.

- 1° Création d'un utilisateur RandoDB
  - Onglet « Général »
    - o Mot de passe : hevs07
    - o Temporaire : TEMP
  - Onglet « Rôle »
    - o Connect
    - o Ressource
  - Onglet « Système »
    - o Create snapshot
    - o Query rewrite
    - o Unlimited tablespace
- 2° Exécution des différents scripts depuis SQLPlus afin de mettre en place un scénario. Les scripts se trouvent sur le DVD dans le dossier « /Base de données/Scripts/ ».
- 3° Ajout des droits d'accès (lecture et exécution, affichage du contenu du dossier et lecture) pour l'utilisateur IUSR\_FREESTEPXP (compte utilisateur IIS) sur le dossier c:/oracle/ora92/. Cette étape est nécessaire pour que le site web puisse accéder à la base de données.

## Installation du site web

Pour installer le site web sur IIS, il suffit de lancer l'exécutable FreeStepSetup.msi présent sur le DVD, dans le dossier « Installation/Setup/ ».

- 1° Site : Site web par défaut
- 2° Virtual directory : FreeStep

## Configuration du site web

### ASP.NET

Après l'installation, il faut vérifier les propriétés du site web « FreeStep » dans IIS. Dans l'onglet ASP.NET, il faut bien faire attention de mettre la version 2.0 d'ASP.NET.

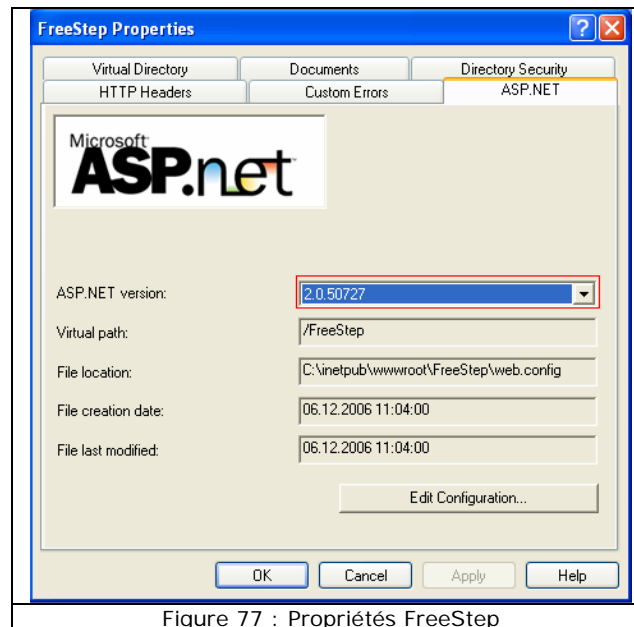


Figure 77 : Propriétés FreeStep

Le bouton « Edit Configuration » permet de modifier quelques valeurs comme la chaîne de connexion à la base de données, le répertoire de destination du site et la clé GoogleMaps.

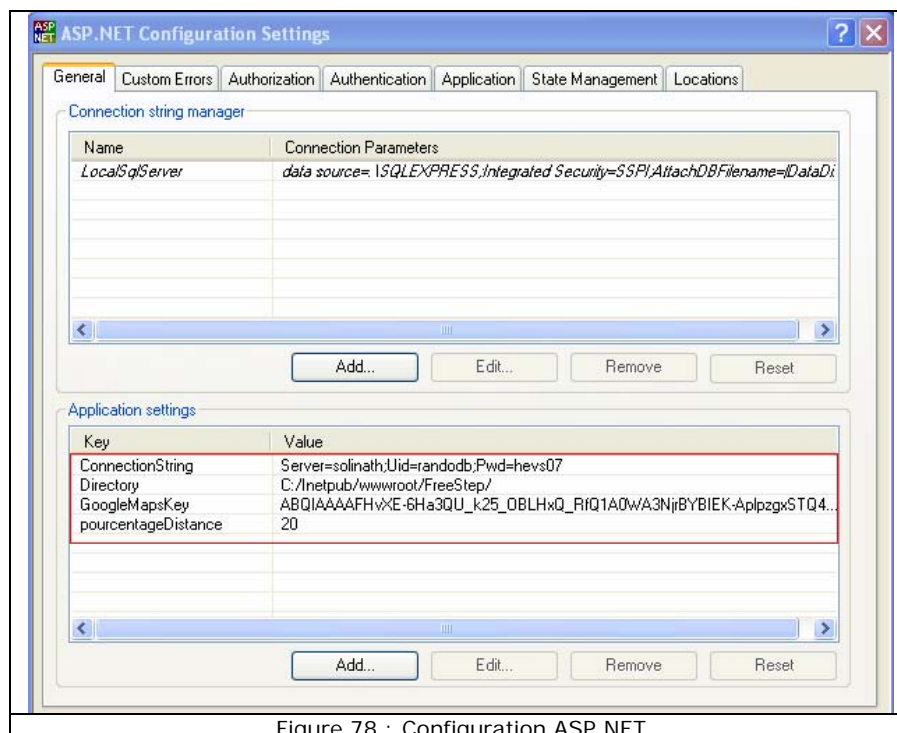


Figure 78 : Configuration ASP.NET



### Droits d'accès

Lors de l'accès à un parcours depuis le site web, la carte GoogleMaps et le fichier GPX sont créés dynamiquement. Afin que cela puisse se faire, il est nécessaire de donner le contrôle total (full control) à l'utilisateur ASPNET sur le dossier c:/inetpub/wwwroot/FreeStep/.

### ChartFX

Pour permettre l'affichage des dénivelés grâce à ChartFX, le dossier « ChartFX70 » (sur le DVD dans le répertoire « Installation/Setup/ ») doit être ajouté dans c:/inetpub/wwwroot/.

Puis, depuis IIS, il suffit d'ajouter un nouveau répertoire virtuel au site web par défaut s'appelant ChartFX70 et pointant sur c:/inetpub/wwwroot/ChartFX70.

Supprimer le nom de l'application dans les propriétés de ChartFX70 (bouton remove)

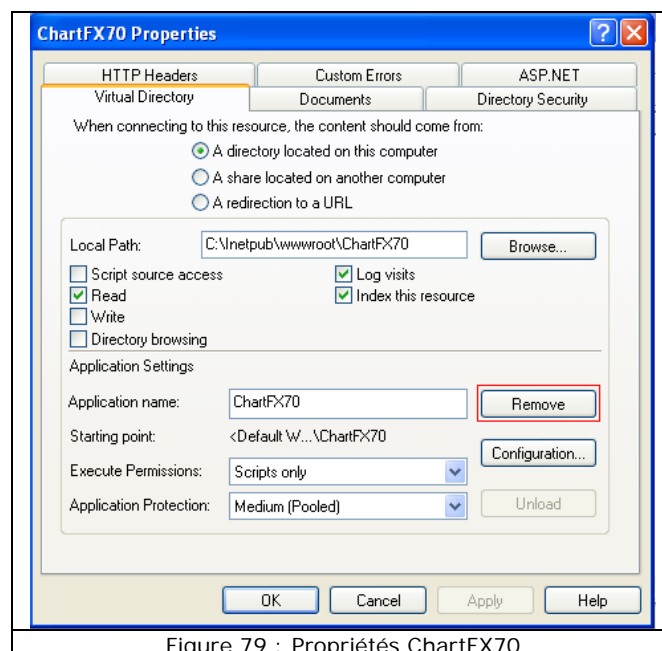


Figure 79 : Propriétés ChartFX70

Finalement, donner tous les droits d'accès aux utilisateurs « IUSR\_FREESTEPXP » et « ASPNET » sur le dossier ChartFx70.

## Description de l'application web

### Page d'accueil

L'accès au site se fait en entrant l'adresse « <http://localhost/FreeStep/> ».

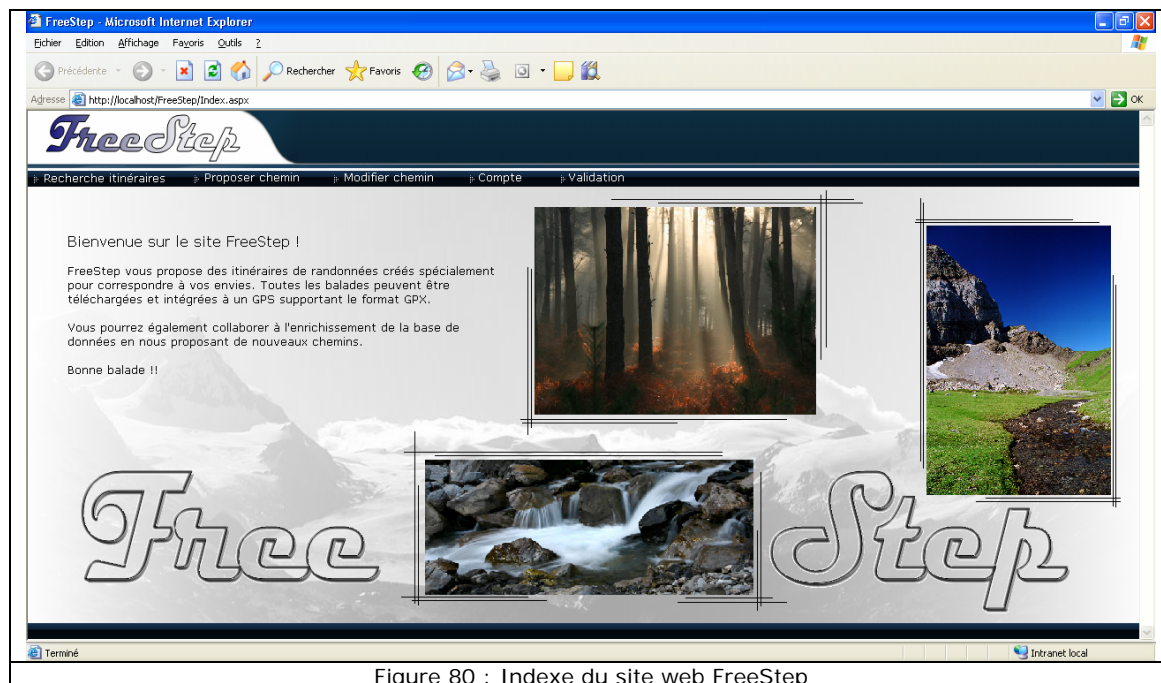


Figure 80 : Index du site web FreeStep

Actuellement, seulement les fonctionnalités de recherche d'itinéraires ont été développées. Il est possible d'y accéder en cliquant soit sur le menu de recherche d'itinéraires « Départ – Arrivée » soit sur le menu « Départ »

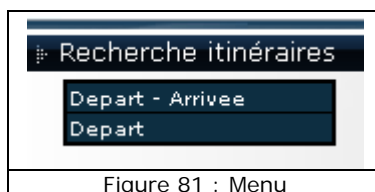


Figure 81 : Menu

## Recherche « Départ – Arrivée »

Cette page permet de faire des recherches d'itinéraires lorsque l'utilisateur connaît son point de départ et son point d'arrivée.

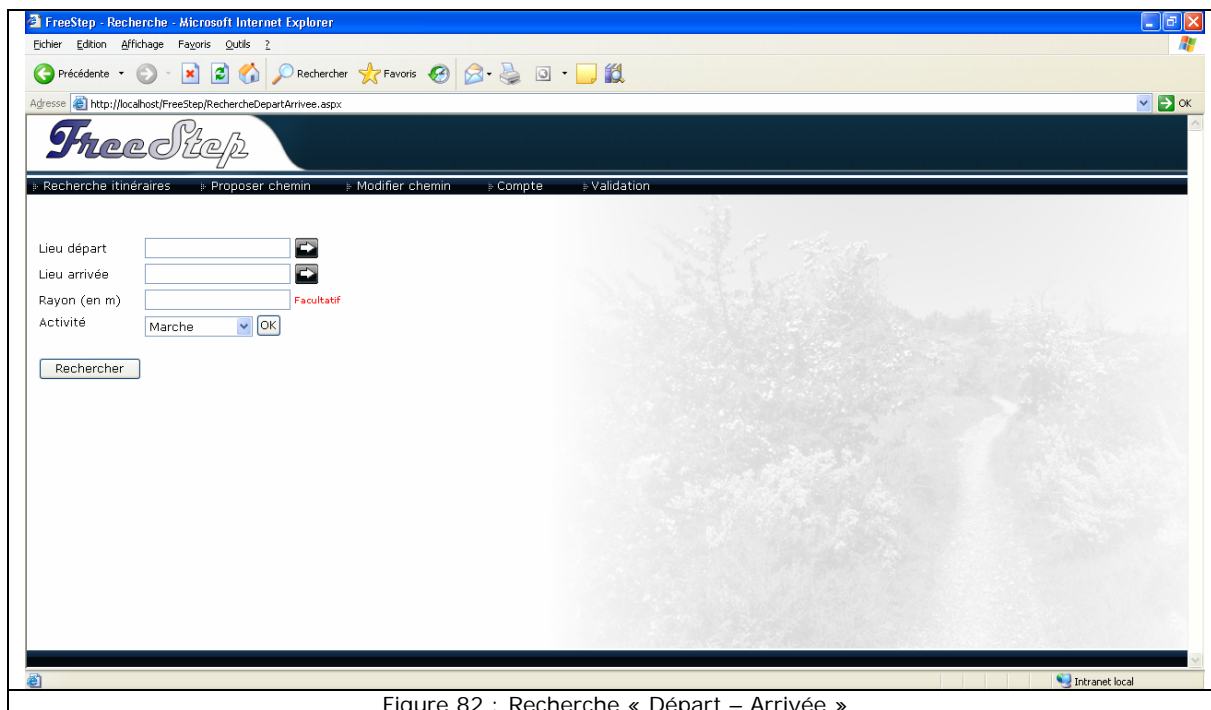



Figure 82 : Recherche « Départ – Arrivée »

En cliquant sur les boutons , l'utilisateur peut faire une recherche sur les lieux de départ et d'arrivée disponibles. Ces lieux peuvent être soit des villes soit des communes. Il est possible d'entrer le nom complet du lieu ou alors, d'entrer une partie du nom suivi d'une étoile « \* ». Par exemple, en entrant « si\* », les lieux « Sierre » et « Sion » seront retournés par l'application.

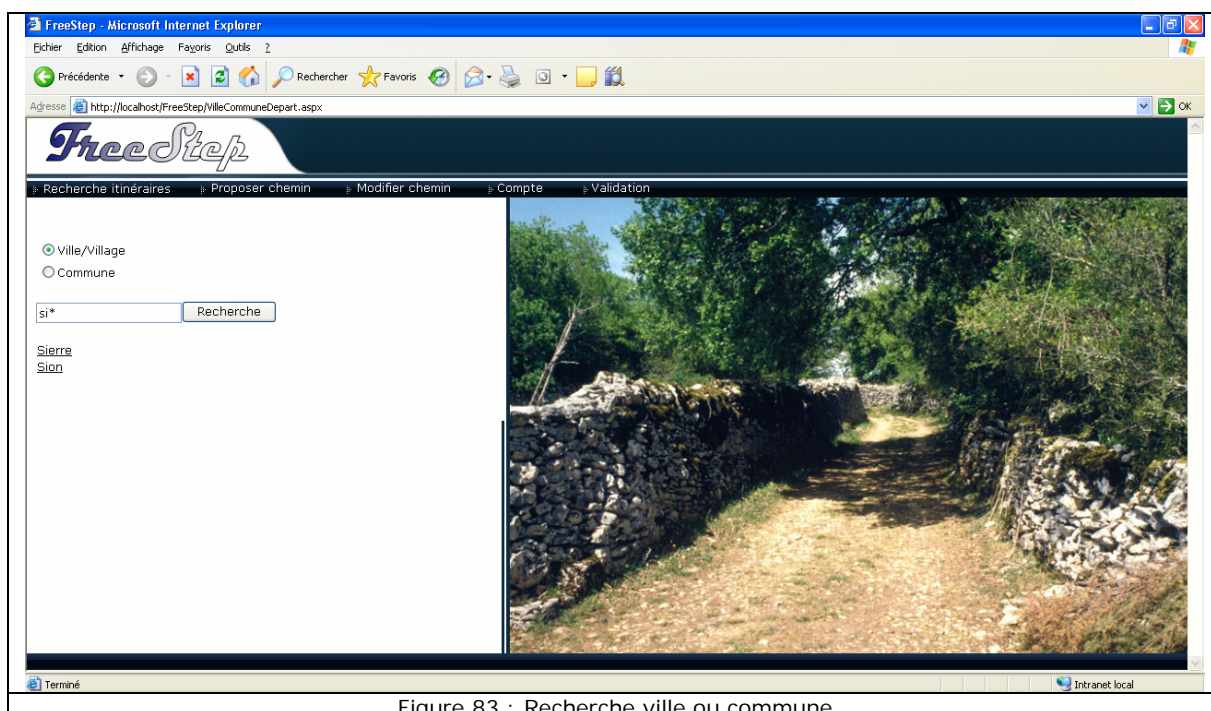


Figure 83 : Recherche ville ou commune

Après avoir choisi les lieux de départ et d'arrivée, l'utilisateur a la possibilité de définir la distance qu'il est d'accord d'effectuer pour rejoindre le départ du parcours (champs « rayon »). Ainsi, si le rayon est de 200, toutes les routes partant dans un rayon de 200 mètres autour du lieu de départ seront prises en considération.

Ensuite, le champ « Activité » permet de choisir quel type d'activité la personne désire effectuer. En cliquant sur le bouton « OK », une liste de type de route correspondant à l'activité choisie est affichée. L'utilisateur peut ainsi décider de n'utiliser que des routes goudronnées et de chemins.

Après avoir cliquer sur le bouton « Rechercher », les résultats s'affichent sur la droite. Des informations sur la distance, la durée et la difficulté de chaque parcours sont indiquées.

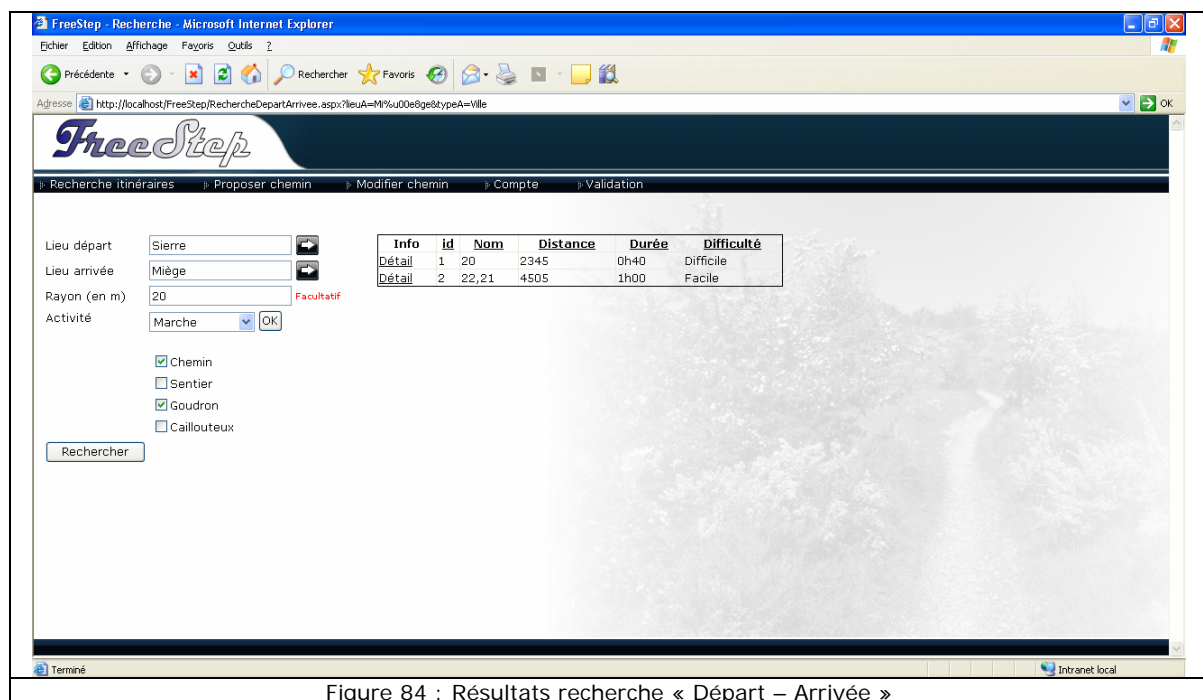


Figure 84 : Résultats recherche « Départ – Arrivée »

## Recherche « Départ »

La recherche « Départ » est utilisée quand une personne connaît son point de départ mais ne désire pas préciser le point d'arrivée.

Cette recherche est similaire à la recherche « Départ – Arrivée » mais possède quelques champs supplémentaires.

En plus du lieu de départ, du rayon, du type d'activité et du type de route, l'utilisateur doit choisir un type de parcours. Il a la possibilité entre un aller simple, un aller-retour ou une boucle.

Le champ « Longueur du trajet » permet de définir la distance maximale des parcours retournés.

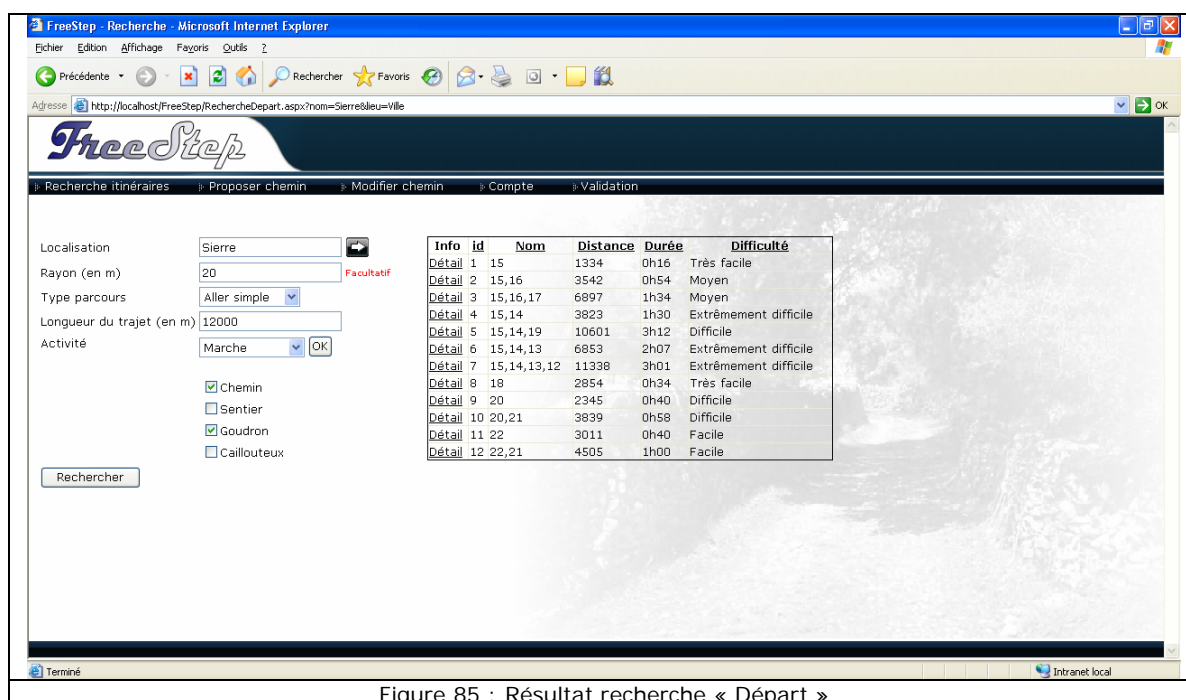


Figure 85 : Résultat recherche « Départ »

A partir du moment où les résultats sont retournés, il est possible de trier les colonnes selon le nom, la distance, la durée ou la difficulté. Pour cela, il suffit de cliquer sur le nom de la colonne à trier.

Info	id	Nom	Distance	Durée	Difficulté
Détail	1	15	1334	0h16	Très facile
Détail	2	15,16	3542	0h54	Moyen
Détail	3	15,16,17	6897	1h34	Moyen

Figure 86 : Résultats

Quand l'utilisateur a choisi le parcours qu'il désire faire, il peut accéder à des informations supplémentaires sur cet itinéraire en cliquant sur Détail.

## Détails

Les informations supplémentaires importantes sont la représentation sur GoogleMap du trajet et des points d'intérêt se situant sur le parcours. Il est possible de cliquer sur les marqueurs présents sur la carte. Cela affichera des informations sur le point d'intérêt sélectionné.

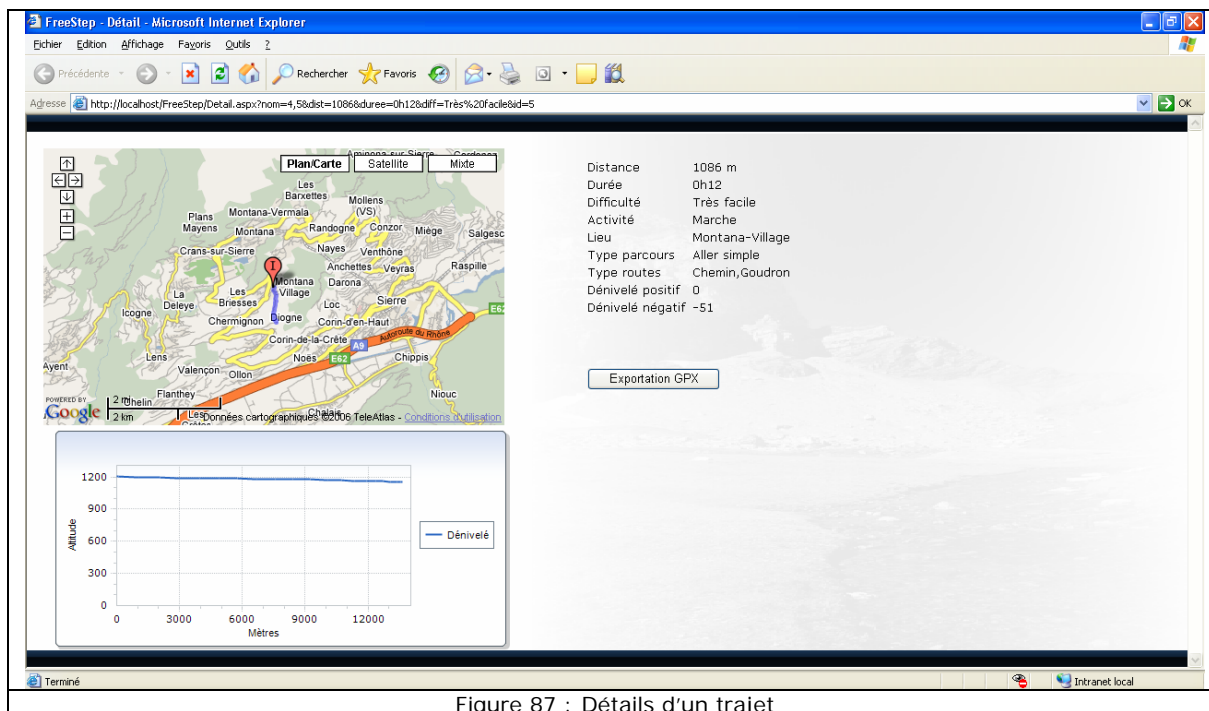


Figure 87 : Détails d'un trajet

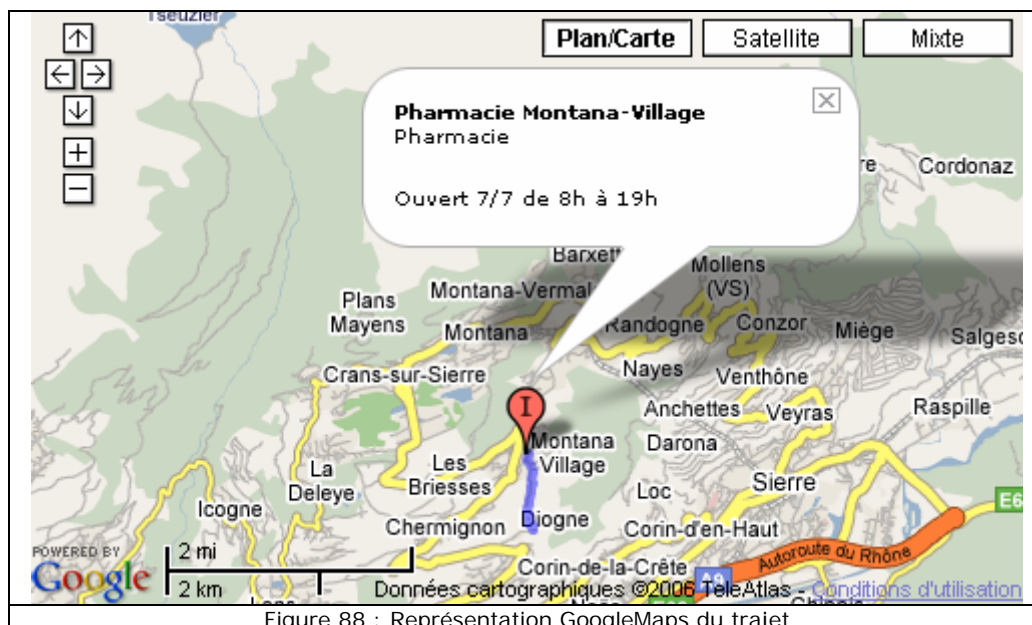


Figure 88 : Représentation GoogleMaps du trajet



En dessous, un graphe représente la dénivellation de l'itinéraire. A droite, le dénivelé positif et le dénivelé négatif donne quelques précisions supplémentaires sur ce graphe.

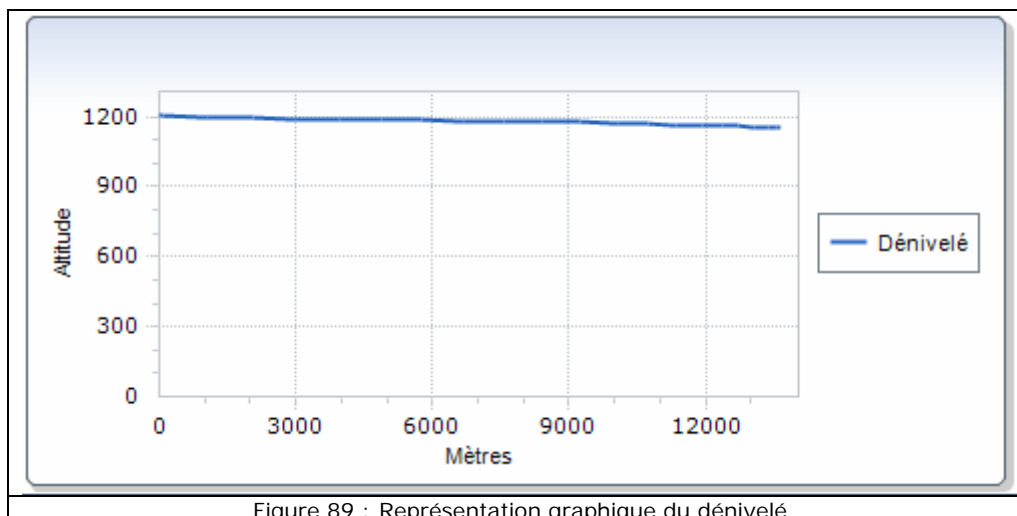


Figure 89 : Représentation graphique du dénivelé

Le bouton « Exportation GPX » crée un fichier GPX contenant les coordonnées du trajet et des points d'intérêt. Ce fichier peut être sauvegardé sur le PC puis intégré dans un GPS compatible avec le format GPX.

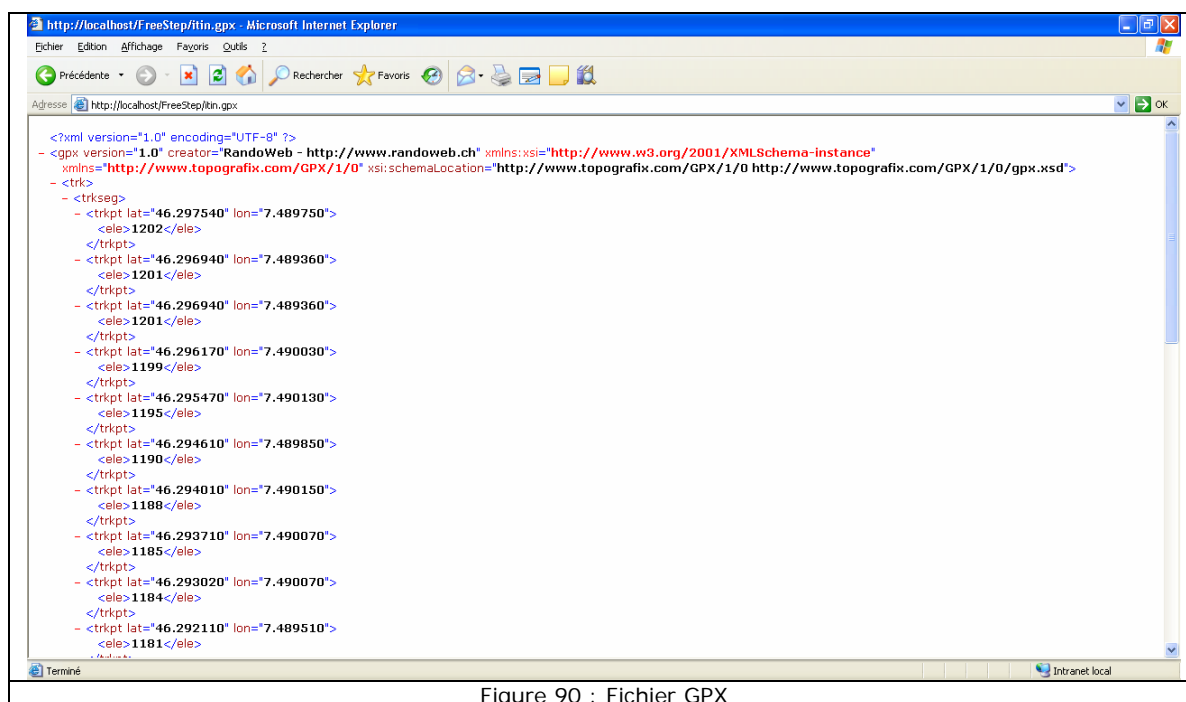


Figure 90 : Fichier GPX

## Création de géométries<sup>37</sup> : Exemples

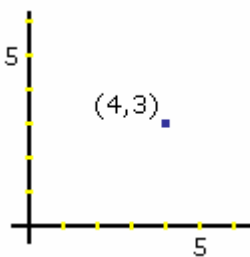
---

### Introduction

Ces exemples présentent des géométries à 2 dimensions. La structure pour une géométrie à 3 dimensions reste cependant la même.

Ces exemples sont représentés sur des axes euclidiens, ce qui permet de laisser la deuxième valeur (srid) à null. Il faudra penser à mettre le système de références spatiales lors de l'utilisation de valeur latitude/longitude.

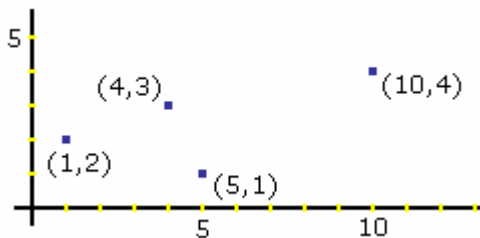
### Point



```
mdsys.sdo_geometry(2001, null, sdo_point_type(4, 3, null), null, null);
```

2001 : Point à 2 dimensions

### Cluster de points



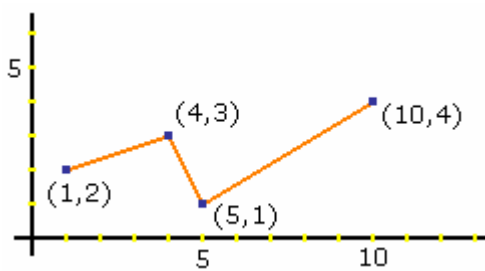
```
mdsys.sdo_geometry(2001, null, null, mdsys.sdo_elem_info_array(1,1,4),
mdsys.sdo_ordinate_array(1,2, 4,3, 5,1, 10,4)) ;
```

2001 : Points à 2 dimensions  
1,1,4 : Commence à la position 1, composé de 4 points (1,4)

<sup>37</sup> Oracle Spatial 9i, 10g (PowerPoint Presentation)  
<http://lbdwww.epfl.ch/e/teaching/SlidesBDA/OracleSpatial.pdf>



## Ligne droite

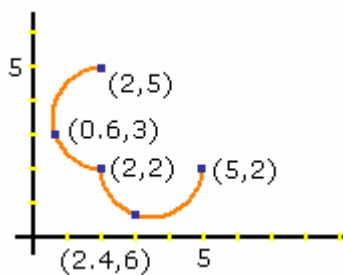


```
mdsys.sdo_geometry(2002, null, null, mdsys.sdo_elem_info_array(1,2,1),
                    mdsys.sdo_ordinate_array(1,2, 4,3, 5,1, 10,4)) ;
```

2002 : Ligne à 2 dimensions

1,2,1 : Commence à la position 1, composé de lignes droites (2,1)

## Arc de cercle

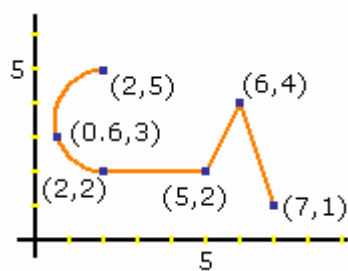


```
mdsys.sdo_geometry(2002, null, null, mdsys.sdo_elem_info_array(1,2,2),
                    mdsys.sdo_ordinate_array(2,5, 0.6,3, 2,2, 2.4,6, 5,2)) ;
```

2002 : Ligne à 2 dimensions

1,2,2 : Commence à la position 1, ligne formée d'arc de cercle (2,2)

## Ligne composée



```
mdsys.sdo_geometry(2002, null, null,
                    mdsys.sdo_elem_info_array(1,4,2, 1,2,2, 5,2,1),
                    mdsys.sdo_ordinate_array(2,5, 0.6,3, 2,2, 5,2, 6,4, 7,1)) ;
```

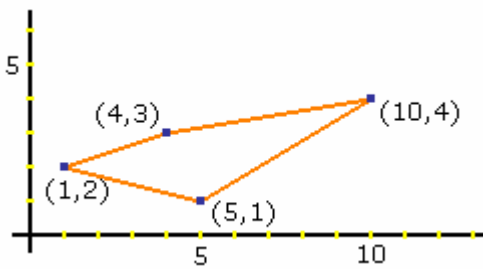
2002 : Ligne à 2 dimensions

1,4,2 : Commence à la position 1, ligne composée de deux éléments

1,2,2 : Commence à la position 1, arc de cercle (2,2)

5,2,1 : Commence à la position 5, ligne droite (2,1)

## Polygone simple avec ligne droite

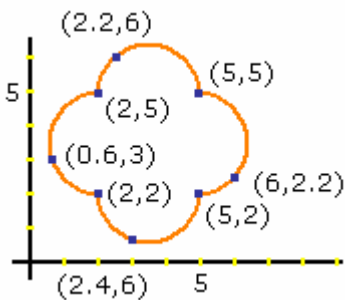


```
mdsys.sdo_geometry(2003, null, null,
  mdsys.sdo_elem_info_array(1,1003,1),
  mdsys.sdo_ordinate_array(1,2, 4,3, 10,4, 5,1, 1,2)) ;

2003 : Polygone à 2 dimensions
1,1003,1 : Commence à la position 1, polygone extérieur (1003) composé
de ligne droite (1)
```

Attention lors de la création de polygone, pour montrer que cette géométrie est fermée, il faut que le premier et le dernier point soient les mêmes.

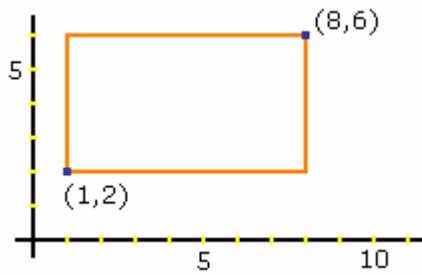
## Polygone simple avec arc



```
mdsys.sdo_geometry(2003, null, null,
  mdsys.sdo_elem_info_array(1,1003,2),
  mdsys.sdo_ordinate_array(2,2, 2.4,6, 5,2, 6,2.2, 5,5, 2.2,6,
    2,5, 0.6,3, 2,2)) ;

2003 : Polygone à 2 dimensions
1,1003,2 : Commence à la position 1, polygone extérieur (1003) composé
d'arcs de cercle (2)
```

## Polygone particulier : rectangle

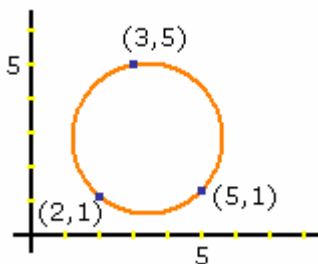


```
mdsys.sdo_geometry(2003, null, null,
                    mdsys.sdo_elem_info_array(1,1003,3),
                    mdsys.sdo_ordinate_array(1,2, 8,6)) ;

2003 : Polygone à 2 dimensions
1,1003,3 : Commence à la position 1, polygone extérieur (1003) rectangle
(3)
```

Pour créer un rectangle, il suffit de 2 points.

## Polygone particulier : cercle

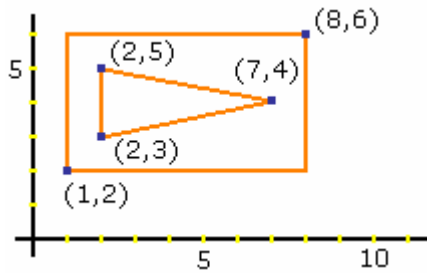


```
mdsys.sdo_geometry(2003, null, null,
                    mdsys.sdo_elem_info_array(1,1003,4),
                    mdsys.sdo_ordinate_array(3,5, 2,1, 5,1)) ;

2003 : Polygone à 2 dimensions
1,1003,4 : Commence à la position 1, polygone extérieur (1003) cercle
(4)
```

Les points doivent être entrés dans le sens inverse des aiguilles d'une montre.

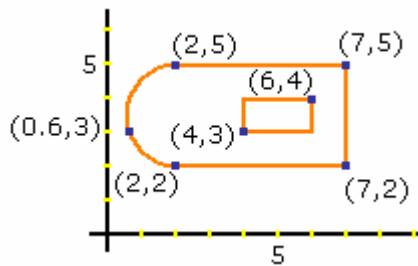
## Polygone composé simple (avec trou et ligne droite)



```
mdsys.sdo_geometry(2003, null, null,
    mdsys.sdo_elem_info_array(1,1003,3, 5,2003,1),
    mdsys.sdo_ordinate_array(1,2, 8,6, 2,3, 7,4, 2,5, 2,3)) ;
```

2003 : Polygone à 2 dimensions  
 1,1003,3 : Commence à la position 1, polygone extérieur (1003) rectangle (3)  
 5,2003,1 : Commence à la position 5, polygone intérieur (2003) composé de ligne droite (1)

## Polygone composé complexe (avec trou, ligne droite et arc de cercle)



```
mdsys.sdo_geometry(2003, null, null,
    mdsys.sdo_elem_info_array(1,1005,2, 1,2,2, 5,2,1, 11,2003,3),
    mdsys.sdo_ordinate_array(2,5, 0.6,3, 2,2, 7,2, 7,5, 2,5, 4,3, 6,4)) ;
```

2003 : Polygone à 2 dimensions  
 1,1005,2 : Commence à la position 1, polygone compose (1005) de deux elements (2)  
 1,2,2 : Commence à la position 1, arc de cercle (2,2)  
 5,2,1 : Commence à la position 5, ligne droite (2,1)  
 11,2003,3 : Commence à la position 11, polygone intérieur (2003) rectangle (3)

## Fonctions et procédures Oracle développées

---

Cette partie présente les différentes fonctions et procédures développées pour ce projet.

### Fonctions

#### *Altitude*

*altitude(geom in mdsys.sdo\_geometry, sens in varchar2) return varchar2*

Retourne sous format textuel, la liste des altitudes d'une géométrie. Selon le sens de la géométrie, le texte retourné sera différent.

Exemple :

```
sdo_geometry(3002, 8307, null, sdo_elem_info_array(1,2,1),
sdo_ordinate_array(46.31,7.58,300, 46.29, 7.53, 100, 46.28, 7.53, 250))
```

si la valeur sens = "True" alors le résultat sera : 300,100,250  
dans le cas contraire, le résultat sera ; 250,100,300

Paramètres :

geom : la géométrie

sens : True si la géométrie est dans le bon sens, False dans le cas contraire

Retour :

la liste sous forme textuelle de toutes les altitudes de la géométrie

#### *DeniveleNeg*

*deniveleNeg(geom in mdsys.sdo\_geometry) return number*

Cette fonction permet de calculer le dénivelé négatif d'une route. Le dénivelé est calculé selon la somme des différences d'altitude lorsque la route descend.

Paramètre :

geom : la géométrie

Retour :

le dénivelé négatif

#### *DenivelePos*

*denivelePos(geom in mdsys.sdo\_geometry) return number*

Cette fonction permet de calculer le dénivelé positif d'une route. Le dénivelé est calculé selon la somme des différences d'altitude lorsque la route monte.

Paramètre :

geom : la géométrie

Retour :

le dénivelé positif

### *DeniveleRap*

*deniveleRap(geom in mdsys.sdo\_geometry) return number*

Cette fonction permet de calculer le rapport d'une route selon son dénivelé positif et la distance parcourue sur l'axe x lorsque la route monte. DeniveleRap est utilisé lorsque la géométrie est dans le bon sens. Le rapport est calculé ainsi :  
dénivelé positif \* 100/distance sur l'axe x

Paramètre :

geom : la géométrie

Retour :

le rapport

### *DeniveleRapNeg*

*deniveleRapNeg(geom in mdsys.sdo\_geometry) return number*

Cette fonction permet de calculer le rapport d'une route selon son dénivelé positif et la distance parcourue sur l'axe x lorsque la route monte. DeniveleRapNeg est utilisé lorsque la géométrie est dans le mauvais sens. Le rapport est calculé ainsi :  
dénivelé positif \* 100/distance sur l'axe x

Paramètre :

geom : la géométrie

Retour :

le rapport

### *Distance*

*distance(geom in mdsys.sdo\_geometry, sens in varchar2) return varchar2*

Retourne sous format textuel, la liste des distances entre chaque point d'une géométrie. Selon le sens de la géométrie, le texte retourné sera différent.

Exemple :

`sdo_geometry(3002, 8307, null, sdo_elem_info_array(1,2,1),  
sdo_ordinate_array(46.31,7.58,300, 46.29, 7.53, 100, 46.28, 7.53, 250))`

si la valeur sens = "True" alors le résultat sera : 0,1345,540  
dans le cas contraire, le résultat sera : 0,540,1345

Paramètres :

geom : la géométrie

sens : True si la géométrie est dans le bon sens, False dans le cas contraire

Retour :

la liste sous forme textuelle de toutes les distances entre chaque point de la géométrie

### *DistanceX*

*distanceX(geom in mdsys.sdo\_geometry) return number*

Cette fonction retourne la somme des distances sur l'axe x lorsque la route monte et qu'elle est dans le bon sens.

Paramètre :

geom : la géométrie

Retour :

la somme des distances sur l'axe x

### *DistanceXNeg*

*distanceXNeg(geom in mdsys.sdo\_geometry) return number*

Cette fonction retourne la somme des distances sur l'axe x lorsque la route monte et qu'elle est dans le mauvais sens.

Paramètre :

geom : la géométrie

Retour :

la somme des distances sur l'axe x

### *Startpoint*

*startPoint(geom in mdsys.sdo\_geometry) return mdsys.sdo\_geometry*

Fonction retournant le point de départ d'une géométrie, c'est-à-dire le premier point entré dans la liste.

Exemple :

`sdo_geometry(3002, 8307, null, sdo_elem_info_array(1,2,1),  
sdo_ordinate_array(46.31,7.58,300, 46.29, 7.53, 100, 46.28, 7.53, 250))`

Le point sera le suivant : `sdo_geometry(3001, 8307,  
sdo_point_type(46.31,7.58,300), null, null)`

Paramètre :

geom : la géométrie

Retour :

une géométrie représentant le premier point de la géométrie

### *TempsDec*

*tempsDec(geom in mdsys.sdo\_geometry, acti in number) return number*

Fonction retournant la durée d'un parcours selon un type d'activité sous format décimal. Cette fonction fait appel à la fonction *deniveleRap* qui lui retourne le rapport du parcours. Grâce à ce rapport et au type d'activité, la fonction recherche la vitesse de la personne. Puis la durée du parcours est calculée en divisant la distance par la vitesse :

$\text{duree} = \text{distance} / \text{vitesse} / 1000$

le résultat sera sous format décimal, comme par exemple 3.75. Ce qui équivaut à 3h45.

Paramètres :

geom : la géométrie

acti : l'identifiant du type d'activité choisie par l'utilisateur

Retour :

la durée du parcours sous format décimal si la route est dans le bon sens

### *TempsDecNeg*

*tempsDecNeg(geom in mdsys.sdo\_geometry, acti in number) return number*

Fonction retournant la durée d'un parcours selon un type d'activité sous format décimal. Cette fonction fait appel à la fonction *deniveleRapNeg* qui lui retourne le rapport du parcours. Grâce à ce rapport et au type d'activité, la fonction recherche la vitesse de la personne. Puis la durée du parcours est calculée en divisant la distance par la vitesse :

$\text{duree} = \text{distance} / \text{vitesse} / 1000$

le résultat sera sous format décimal, comme par exemple 3.75. Ce qui équivaut à 3h45.

Paramètres :

geom : la géométrie

acti : l'identifiant du type d'activité choisie par l'utilisateur

Retour :

la durée du parcours sous format décimal si la route est dans le mauvais sens



## Package

Le package Geom a été créé car il rassemble des procédures et des fonctions renvoyant ou utilisant des curseurs. Comme une procédure ne peut pas retourner directement un curseur, il faut ajouter au corps du package un type de données personnalisé correspondant à une référence du curseur (ref cursor).

### *RechActivite*

```
rechActivite(dataOut out t_cursor)
```

Procédure retournant la liste des identifiants et des noms des activités présentes dans la base de données.

Retour :  
liste des activités présentes dans la base de données

### *RechTypeRoute*

```
rechTypeRoute(text in varchar2, dataOut out t_cursor)
```

Procédure retournant l'identifiant et le nom des type de routes correspondant à une ou plusieurs activités.

Paramètre :  
text : activité

Retour :  
liste des types de route

### *RechDifficulte*

```
rechDifficulte (dataOut out t_cursor)
```

Procédure retournant le type, le rapport minimal et le rapport maximal de chaque difficulté

Retour :  
toutes les difficultés

### *RechVille*

```
rechVille(text in varchar2, dataOut out t_cursor)
```

Retourne une liste de ville correspondant au texte entré.

Exemple :

« Sion » retournera « Sion »  
« Si\* » retournera « Sion » et « Sierre »

Paramètre :

text : texte

Retour :

liste de ville

### *RechCommune*

```
rechCommune(text in varchar2, dataOut out t_cursor)
```

Retourne une liste des communes correspondant au texte entré.

Exemple :

« Sion » retournera « Sion »  
« Si\* » retournera « Sion » et « Sierre »

Paramètre :

text : texte

Retour :

liste des communes

### *DistanceDepartArrivee*

```
distanceDepartArrivee(lieuD in varchar2, typeLieuD in varchar2, lieuA in  
varchar2, typeLieuA in varchar2, distance out t_cursor)
```

Procédure retournant la distance entre deux lieux.

Paramètres :

lieuD : lieu de départ du trajet  
typeLieuD : « Ville » si lieuD est une ville, « Commune » si c'est une commune  
lieuA : lieu d'arrivée du trajet  
typeLieuA : « Ville » si lieuA est une ville, « Commune » si c'est une commune

Retour :

la distance entre les deux lieux

### ZoneBuffer

```
zoneBuffer(lieu in varchar2, typeLieu in varchar2, dist in number) return
mdsys.sdo_geometry
```

Fonction retournant une géométrie correspondant à une zone autour du lieu de distance « dist ». Si le lieu est une ville, la géométrie retournée sera un cercle autour de cette ville, si c'est une commune, la géométrie sera un polygone arrondi.

Paramètres :

lieu : nom du lieu

typeLieu : « Ville » si lieu est une ville, « Commune » si c'est une commune

dist : longueur en mètre de la distance autour du lieu

Retour :

géométrie correspondant à une zone buffer autour d'un lieu

### Requêtes Visual Studio

Ces requêtes sont utilisées dans Visual Studio. Il n'était pas pratique de passer ces fonctions par Oracle car des champs devaient être traités par Visual Studio pour pouvoir être utilisés.

#### SelectDepart

```
selectDepart(string lieu, string typeLieu, string rayon, string route)
```

Requête retournant la liste des identifiants des routes qui se trouvent dans la zone Buffer autour de « lieu » et de distance « rayon ». Elle permet donc de sélectionner toutes les routes de départ.

Paramètres :

lieu : nom de la localité ou de la commune de départ

typeLieu : 'Ville' ou 'Commune'

rayon : distance autour de "lieu" pour la début de l'itinéraire

route : liste des types de routes autorisés pour ce parcours

#### SelectDistance

```
selectDistance(string lieu, string typeLieu, string distance, string
route, string activite)
```

Cette requête retourne la longueur du parcours, le dénivelé positif, le dénivelé négatif, la durée du parcours sous format décimal dans les deux sens de la route, la distance sur l'axe x dans les deux sens et les coordonnées du point de départ de chaque route.

Paramètres :

lieu : nom de la localité ou de la commune de départ

typeLieu : 'Ville' ou 'Commune'

distance : longueur du parcours

route : liste des types de routes autorisée pour ce parcours

### *SelectLien*

```
selectLien(string lieu, string typeLieu, string rayon, string distance,  
string route)
```

Retourne la liste des liens entre les routes se trouvant dans la zone Buffer. On trouve les identifiants ainsi que les points d'intersection entre ces routes. Cette requête est en fait une union de 3 requêtes. La première et la deuxième retourne la liste des routes ayant un lien avec les routes de départ. La troisième retourne les liens entre les routes externes.

Paramètres :

- lieu : nom de la localité ou de la commune de départ
- typeLieu : 'Ville' ou 'Commune'
- rayon : distance autour de "lieu" pour la début de l'itinéraire
- distance : longueur du parcours
- route : liste des types de routes autorisés pour ce parcours

### *SelectPointsRoute*

```
selectPointsRoute(string identifiant)
```

Sélectionne les coordonnées de tous les points des routes désirées.

Paramètre :

- identifiant : identifiants des routes pour lesquelles il faut chercher les points.

### *SelectPointsInteret*

```
selectPointsInteret(string identifiant)
```

Sélectionne la liste des points d'intérêts se trouvant dans une zone de 50 mètres autour du trajet trouvé.

Paramètre :

- identifiant : identifiants des routes

### *SelectDenivele*

```
selectDenivele(string identifiant, File f)
```

Requête retournant la liste des altitudes et des distances permettant l'affichage graphique du dénivelé.

Paramètres :

- identifiant : identifiants des routes
- f : File correspondant au trajet

## Problèmes rencontrés

### Oracle Spatial

*Arc\_tolerance*<sup>38</sup> (ora-13292)

```
SQL> @buffer
select mdsys.sdo_geom.sdo_buffer(a.position, 100, 1.2)
*
ERREUR à la ligne 1 :
ORA-13292: Spécification ARC_TOLERANCE incorrecte
ORA-06512: à "MDSYS.SDO_SGL", ligne 439
ORA-06512: à "MDSYS.SDO_GEOM", ligne 3022
ORA-06512: à "MDSYS.SDO_GEOM", ligne 3034
```

Figure 91 : Erreur arc\_tolerance

Cette erreur apparaît lors de l'utilisation de `sdo_buffer`, `sdo_aggr_buffer` ou `sdo_arc_densify` sur une géométrie géodésique. Pour corriger ce problème, `arc_tolerance` doit être spécifié et il ne doit pas être plus petit que la tolérance spécifiée pour la géométrie.

Il faut aussi spécifier les nombres à virgules avec des virgules et non par des points : 0,05 et non 0.05

```
SQL> SELECT mdsys.sdo_geom.sdo_buffer(a.position, b.diminfo, 1, 'unit=km arc_tolerance=0,05') from
ville a, user_sdo_geom_metadata b where table_name = 'VILLE' and a.id=1 ;

MDSYS.SDO_GEOM.SDO_BUFFER(A.POSITION,B.DIMINFO,1,'UNIT=KMARC_TOLERANCE=0,05')(SD
-----
SDO_GEOMETRY(3003, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1), SDO_ORDINATE_ARR
AY(46,233, 7,35904221, 0, 46,233, 7,34095779, 0, 46,233, 7,35904221, 0))
```

Figure 92 : Valeur arc\_tolerance

*Création d'index* (ora-29855)

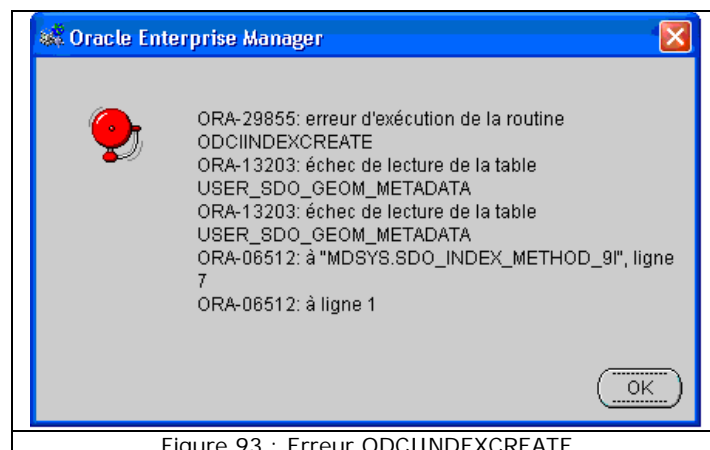


Figure 93 : Erreur ODCIINDEXCREATE

Lors de la création d'index, cette erreur peut apparaître si les métadonnées de la table correspondante n'ont pas été créées. Il faut faire attention de créer les métadonnées avant les index.

<sup>38</sup> <http://forums.oracle.com/forums/thread.jspa?threadID=289425>

*Dim\_array (ora-29875)*

```
SQL> @insert
insert into route values(1, 'Sierre - Sion', mdsys.sdo_geometry(3002, 8307, null, mdsys.sdo_elem_inf
*
ERREUR à la ligne 1 :
ORA-29875: échec d'exécution de la routine ODCIINDEXINSERT
ORA-13364: Les dimensions de couche ne correspondent aux dimensions de
géométrie
ORA-06512: à "MDSYS.SDO_INDEX_METHOD_9I", ligne 319
ORA-06512: à ligne 1
```

Figure 94 : Erreur ODCIINDEXINSERT

Lors de l'insertion dans une table géométrique d'une géométrie, il faut bien faire attention aux dimensions de l'objet inséré. Si, dans les métadonnées, cette colonne a été définie comme étant en 2 dimensions, il n'est pas possible d'insérer des données à 3 dimensions.

Pour corriger le problème, il faut modifier l'insertion ou les métadonnées selon les besoins.

*Retour de réponse incorrect*

Pour que les données géométriques soient correctes, il faut prendre garde au format des données qu'on insère. L'exemple ci-dessous va illustrer le problème :

```
SDO_GEOMETRY(3003, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1),
SDO_ORDINATE_ARRAY(46,0999979, 7,54204213, 0, 46,0909391, 7,53299784, 0,
46,1000021, 7,52395786,0, 46,2999979, 7,52395786, 0, 46,3090609, 7,53299784,
0, 46,3000021, 7,54204213, 0, 46,0999979, 7,54204213, 0))
```

N'est pas similaire à :

```
SDO_GEOMETRY(3003, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1),
SDO_ORDINATE_ARRAY(46.0999979, 7.54204213, 0, 46.0909391, 7.53299784, 0,
46.1000021, 7.52395786,0, 46.2999979, 7.52395786, 0, 46.3090609, 7.53299784,
0, 46.3000021, 7.54204213, 0, 46.0999979, 7.54204213, 0))
```

Dans le premier exemple, les latitudes et longitudes sont exprimée avec des virgules (,), alors que dans le deuxième, avec des points (.).

L'exemple correct est le deuxième car dans le premier, Oracle va croire qu'entre chaque virgule, il y a une valeur.

*Fonctions géométriques sur des géométries*

Avec Oracle Spatial, il n'est pas possible d'utiliser des fonctions géométriques sur des géométries n'ayant pas d'index. Il fallait donc créer des « fonction-based spatial index », c'est-à-dire un index sur la fonction renvoyant une géométrie. Ce point est détaillé dans le chapitre « Utilisation d'Oracle Spatial ».

### Altitude

Les fonctions géométriques proposées par Oracle Spatial ne gèrent pas l'altitude.

```
SQL> Select trace from chemin where id in(1,2) ;

TRACE(SDO_GTYPE, SDO_SRID, SDO_POINT(X, Y, Z), SDO_ELEM_INFO, SDO_ORDINATES)
-----
SDO_GEOMETRY(3002, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 2, 1), SDO_ORDINATE_ARRAY(
46,1, 7,533, 100, 46,3, 7,533, 100))

SDO_GEOMETRY(3002, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 2, 1), SDO_ORDINATE_ARRAY(
46,291, 7,533, 100, 46,3, 7,533, 100))

SQL> Select mdsys.sdo_geom.sdo_union(c1.trace, c2.trace,1) from chemin c1, chemin c2
2 where c1.id = 1 and c2.id = 2 ;

MDSYS.SDO_GEOM.SDO_UNION(C1.TRACE,C2.TRACE,1)(SDO_GTYPE, SDO_SRID, SDO_POINT(X,
SDO_GEOMETRY(3002, 8307, NULL, SDO_ELEM_INFO_ARRAY(1, 2, 1), SDO_ORDINATE_ARRAY(
46,1, 7,533, 0, 46,3, 7,533, 0, 46,291, 7,533, 0))
```

Figure 95 : Gestion de l'altitude

L'exemple ci-dessus crée (union) une LineString avec 2 LineString. On peut voir que les longitude et latitude ont été liées correctement, cependant l'altitude n'est pas gérée. Elle est même transformée en 0. Il est donc obligatoire de gérer manuellement l'altitude de chaque point d'une géométrie.

### Fonction relate - disjoint

La fonction `sdo_geom.relate()` permet de connaître les relations existantes entre des géométries.

Lors de la recherche d'itinéraire, il était important de connaître les routes n'étant pas en relation avec la zone contenant les routes de départ (cf : zone rouge). Pour cela, Oracle Spatial propose d'utiliser la fonction `sdo_geom.relate()` avec l'attribut « disjoint », comme le montre l'image ci-dessous.

```
SQL> SELECT id FROM chemin
2 WHERE mdsys.sdo_relate(trace, (select types.zonebuffer('Sierre', 'Ville', 20) from dual), 'mask
=disjoint querytype=WINDOW') = 'TRUE';

ID
-----
12
```

Figure 96 : Requête relate avec disjoint

Afin de vérifier le résultat, une autre requête retournant le type de relation entre chaque route et la zone buffer a été développée. Les résultats suivants ont été obtenus :

ID	RELATION
12	DISJOINT
1	OVERLAPBDYDISJOINT
11	OVERLAPBDYDISJOINT
2	OVERLAPBDYDISJOINT
4	OVERLAPBDYDISJOINT
5	OVERLAPBDYDISJOINT
6	DISJOINT
10	DISJOINT

Figure 97 : Requête relate

On peut voir ici, qu'en réalité, il y a trois routes étant « disjoint » à la zone buffer.

Pour corriger ce problème, la seule solution trouvée, a été de modifier la requête ainsi :

```
SQL> SELECT id FROM chemin
2  WHERE mdsys.sdo_geom.relate(trace, 'determine', (select types.zonebuffer('Sierre', 'Ville', 20)
from dual), 0.05) = 'DISJOINT';
```

ID
12
6
10

Figure 98 : Requête modifiée

## Visual Studio

### Affichage dans une GridView

Pour afficher les différents itinéraires trouvés, une DataTable doit être créée. Toutes les informations des trajets sont ensuite ajoutées à cette DataTable puis celle-ci est liée à une GridView.

Actuellement, les informations sont affichées comme dans l'image ci-dessous.

Info	id	Nom	Distance	Durée	Difficulté
Détail	2	2	993	0h12	Facile

Figure 99 : GridView

Il serait intéressant de pouvoir cacher certaines colonnes, comme la « id », car elle n'est pas utile à l'utilisateur. Il ne faut pas supprimer la colonne, car il est nécessaire d'avoir toutes les informations pour pouvoir par la suite afficher les détails sur ce trajet.

Cependant, il n'est pas possible de cacher les colonnes lorsque la source de données d'une GridView est liée dans le code et non depuis le designer.

Par exemple, si l'on décide de cacher la colonne 1 (= id), l'erreur suivante apparaît :

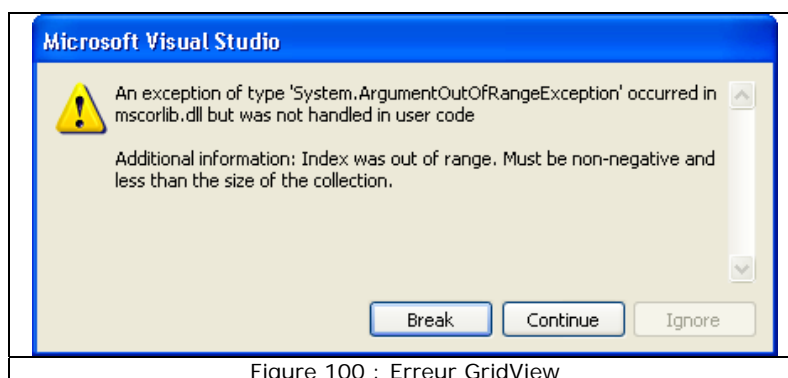


Figure 100 : Erreur GridView

Cela signifie que la GridView ne possède pas de colonne 1.

Différentes recherches sur Internet ont été faites, mais aucune solution n'a encore été trouvée.



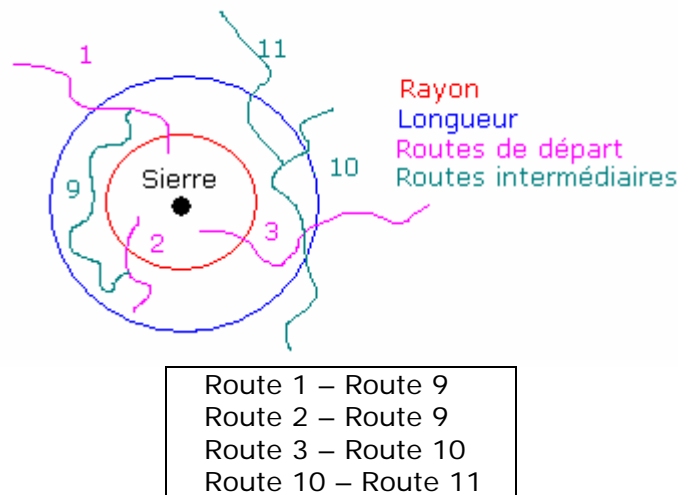
## Graphe

### Recherche d'itinéraires dynamique

Beaucoup de recherches et de tests ont été fait afin de trouver des itinéraires de façon dynamique mais à chaque fois, un problème se présentait.

Il aurait été intéressant de pouvoir créer les itinéraires depuis une requête plutôt qu'au niveau programmation C# car ainsi l'utilisation des fonctionnalités géométriques d'Oracle Spatial était possible.

En premier lieu, une requête renvoyant les liens des routes entre elles a été développée :



Ce qui veut dire, que les itinéraires suivants pouvaient être retournés :

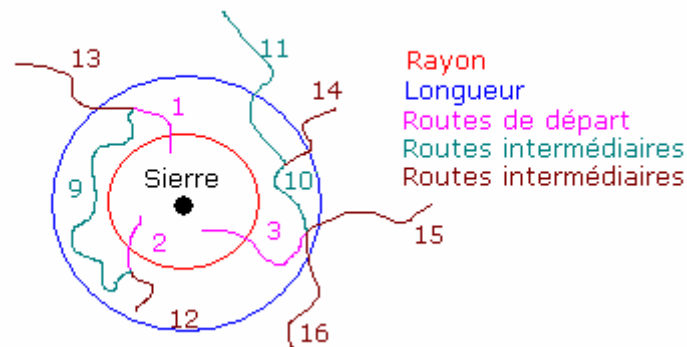
Route 1 – Route 9
Route 2 – Route 9
Route 1 – Route 9 – Route 2
Route 3 – Route 10
Route 3 – Route 10 – Route 11

Le premier problème a été le fait que les routes pouvaient être « coupées » par une autre en plein milieu. Il fallait donc connaître la distance entre 2 points d'une LineString. Cette fonction a été facilement développée, mais on peut rapidement se rendre compte, que connaître les deux points pour calculer cette distance était plus difficile.

Par exemple, si on prend les chemin 3 – 10 – 11, il n'est pas trop compliqué de trouver la distance entre 3 et 10 car sur la route 3, le premier point est le point de départ ou de fin de la LineString et le deuxième point est le point d'intersection des 2 droites. Pour le chemin 10 – 11, c'est le point d'intersection de 3 et 10 ainsi que le point d'intersection de 10 et 11. Et finalement, c'est la distance de la route 11.

Comme il faut prendre en considération plusieurs critères différents (startpoint, endpoint, point d'intersection ou longueur de la géométrie), il n'est pas vraiment envisageable de le faire avec un requête.

Pour rendre les recherches plus faciles, la représentation des routes a été modifiée. Ainsi, chaque croisement correspondrait à la fin d'une route et au début d'une autre.



Pour calculer la distance 3 – 10 – 11, il suffit de faire la somme des 3 distances.

Après réflexion, on se rend compte qu'il est nécessaire d'utiliser de la récursivité pour accéder aux différentes routes. Il n'était donc pas possible de retourner l'ensemble des itinéraires possibles en une requête. Plusieurs idées ont été envisagées pour résoudre ce problème. La première a été d'utiliser de « Global Temporary Table ». Cependant, ces tables temporaires n'acceptent pas des champs de type `sdo_geometry`.

Puis, l'idée de l'emploi de « Cursor » imbriqués a été envisagé mais ne connaissant pas le nombre total de routes qui se suivent, il n'est pas envisageable de le faire de cette façon.

Finalement, ne pouvant pas réellement faire la totalité du travail depuis Oracle, il a été décidé de faire la recherche d'itinéraire grâce à des Graphes sur une plateforme Visual Studio C#.

## Recherche « Départ – Arrivée »

Adresse: <http://localhost:4772/RechercheDepartArrivee.aspx?lieuA=Bern&typeA=Ville>

Recherche itinéraires | Proposer itinéraire | Modifier itinéraire | Compte | Validation

Lieu départ: Sierre  
 Lieu arrivée: Bern  
 Rayon (en m): 20  
 Activité: Marche

Id	id	nom	distance	temps	rapport
4,10	1	4,10	98159	19h37	Facile
5	2	5	68490	13h41	Facile
5,10	3	5,10	145102	32h50	Facile

Facultatif

Editeur de table : "TEST"."CHEMIN" - sys@SOLINATH

ID	NOM	IDTYPEROUTE
12	Salgesch - chippis	2
1	Sierre - La Sage	2
11	Salgesch - Sierre	2
2	Sierre - Chippis	2
4	Sierre - Sion	2
5	Sierre - Bern	2
10	Sion - Bern	2

Rech

Figure 101 : Erreur recherche « Départ – Arrivée »

Cette image montre les résultats d'une recherche d'itinéraires entre Sierre et Berne. Les deux premières lignes sont correctes mais la troisième propose un itinéraire passant par Sierre-Bern / Bern-Sion. On arriverait donc à Sion et non à Berne. Cette erreur s'explique par le fait que les routes Sion-Bern et Sierre-Bern sont des routes d'arrivée. Par conséquent, le parcours 5-10 est composé de deux routes d'arrivée et était jusqu'à présent considéré comme correct car finissant par une route d'arrivée. En réalité, il ne faut pas prendre en considération les trajets dont les deux derniers tronçons sont des routes d'arrivée (cas se produisant lorsqu'il y a une boucle). Il suffit donc de tester si l'avant dernier Noeud de la File est une route d'arrivée et dans ce cas, éliminer le parcours.

## IIS - Oracle

## Client Oracle

Lors du transfert de l'application web sur IIS, une erreur « System.Data.OracleClient requires Oracle client software version 8.1.7 or greater » est apparue.

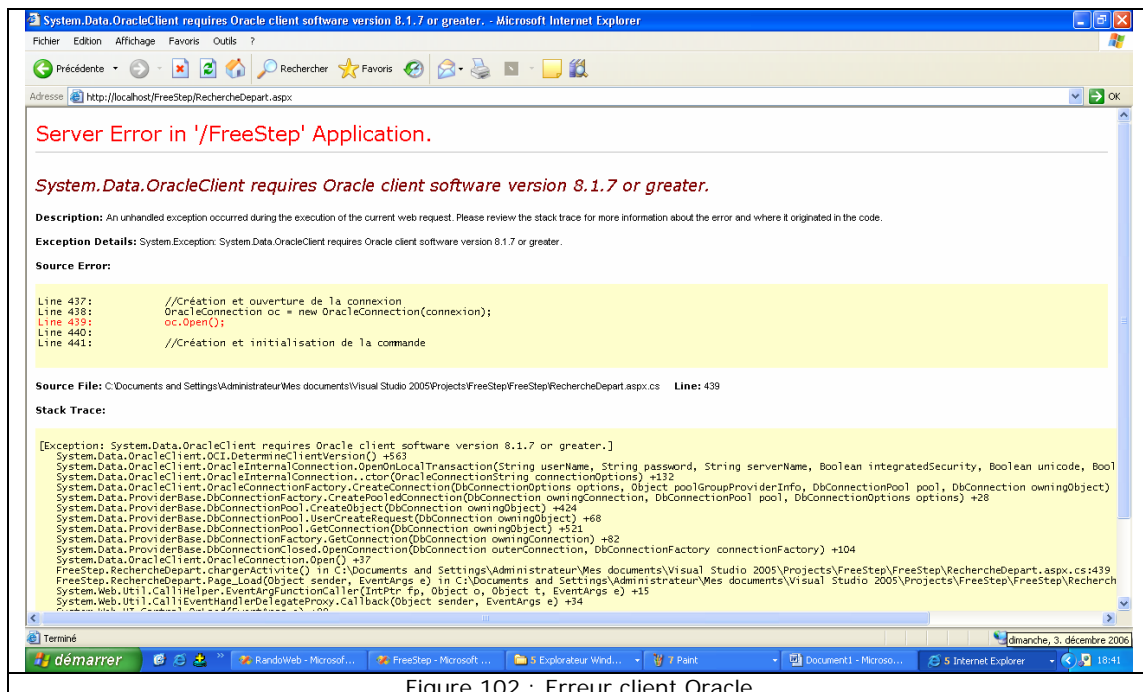


Figure 102 : Erreur client Oracle

Oracle client Runtime a donc été installé. Mais ça n'était pas suffisant pour supprimer l'erreur. Après quelques recherches sur Internet, quelques informations sur <http://dotnetjunkies.com/WebLog/rtgurskevik/archive/2005/01/19/45958.aspx> ont été trouvées où il était signalé que l'erreur provenait d'un problème de droits d'accès. Après avoir suivi les instructions proposées, cette erreur a été corrigée. Cependant, une autre est apparue :

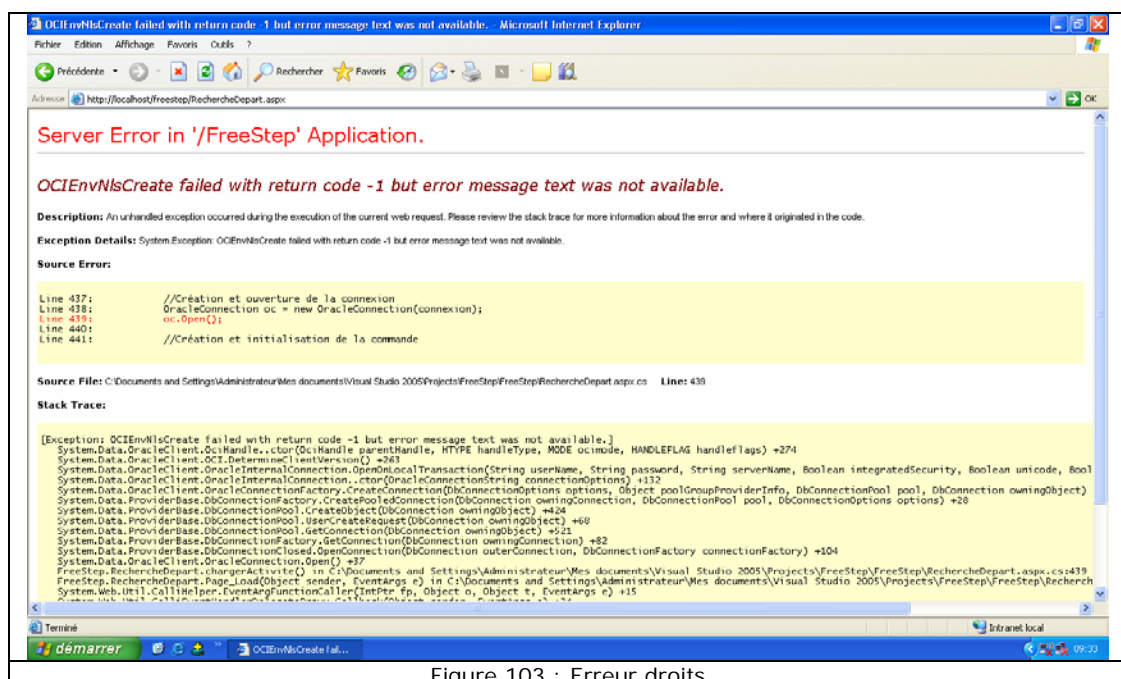


Figure 103 : Erreur droits

Pour corriger cette erreur, il suffit de rajouter pour le dossier ora92 les droits « lectures et exécution, affichage et lecture » à l'utilisateur IIS (IUSR\_NAT)

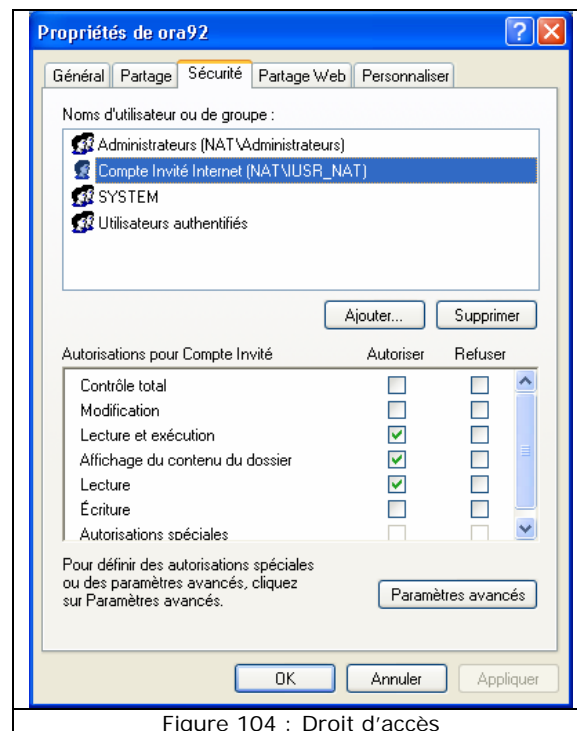


Figure 104 : Droit d'accès

### Création de fichier depuis IIS

Sur IIS, pour pouvoir créer un fichier depuis l'application web dans le dossier du site, il faut donner tous les droits à l'utilisateur ASP.NET sur ce dossier.

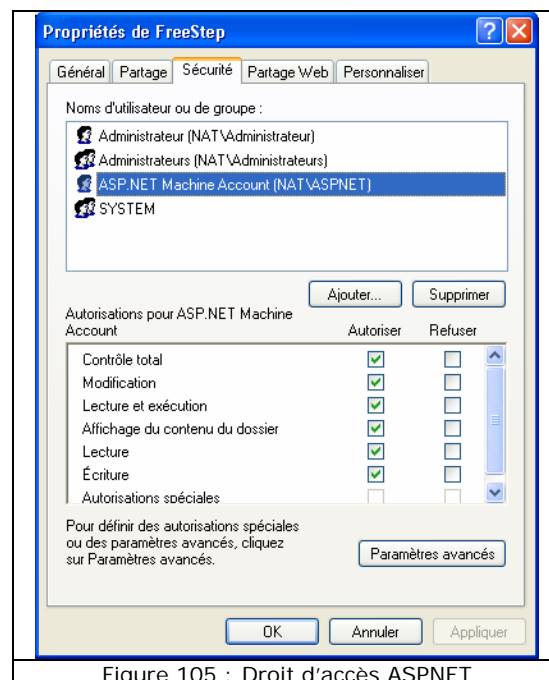


Figure 105 : Droit d'accès ASPNET

## Personnes de contact

---

### **Pierre Gonin**

Inser SA  
Chemin de Maillefer 36  
1052 Le Mont-sur-Lausanne  
021/643 77 11  
[pg@inser.ch](mailto:pg@inser.ch)

### **Yves Molk**

Etat du Valais  
Service des routes et cours d'eau  
Géoroute  
Rue des Creusets 5  
1950 Sion  
027/606 34 21  
[Yves.molk@admin.vs.ch](mailto:Yves.molk@admin.vs.ch)

### **Rainer Oggier**

Etat du Valais  
Service des mensurations cadastrales et de la géomatique  
Chef du CC GEO  
Avenue de la Gare 39  
1950 Sion  
027/606 28 28  
[Rainer.oggier@admin.vs.ch](mailto:Rainer.oggier@admin.vs.ch)

### **Serge Gaudin**

Etat du Valais  
Service des routes et cours d'eau  
Commission cantonale de signalisation routière  
Route des Iles  
Les Ronquoz  
1950 Sion  
027/606 36 56  
[Serge.gaudin@admin.vs.ch](mailto:Serge.gaudin@admin.vs.ch)

Toutes ces personnes ont été rencontrées ou contactées lors de la recherche de données existantes à intégrer dans la base de données.


## Contenu du DVD

**Filière informatique de gestion**

**Diplôme 2005 / 2006**

**Nathalie Solioz**

**Intégration des données spatiales  
dans les bases de données courantes**



JavaScript Tree Menu

- Contenu
  - Documents
    - Rapport final
    - Rapports hebdomadaires
    - Planification
    - PV
    - PDM et use cases
    - Exemples
    - Données
    - Sources externes
  - Sources
    - Site internet
  - Base de données
    - Scripts
  - Installation
    - Document d'installation
    - Setup
  - Outils
    - Base de données
    - Gestion des données
    - Visual Studio
    - Web Application

**Documents**  
 Dans cette partie du DVD, vous trouverez l'ensemble des documents créés ou utilisés durant ce travail de diplôme  
*Rapports, planification, PV, PDM et use cases, sources externes, ...*

**Sources**  
 Ce dossier comporte l'ensemble des sources du site Internet

**Base de données**  
 Afin de pouvoir recréer entièrement la partie pratique de ce projet, les différents scripts de création de la base de données sont fournis

**Installation**  
 Ici vous trouverez le setup d'installation du site web ainsi qu'un document d'installation

**Outils**  
 Tous les outils utilisés durant ce projet sont répertoriés dans ce dossier  
*Base de données, WebApplication, ChartFX, ...*

## Attestation

---

Je déclare, par ce document, que j'ai effectué le travail de diplôme ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de diplôme, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

Professeur Yann Bocchi

Sierre, le 11 décembre 2006

Solioz Nathalie